CRANFIELD UNIVERSITY

DERECK D WEBSTER

AUTOMATIC RAIN DROP DETECTION FOR IMPROVED SENSING IN AUTOMOTIVE COMPUTER VISION APPLICATIONS

SCHOOL OF ENGINEERING

MSc THESIS

2013-2014

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

Advanced Mathematics and Computing

(Digital Signal and Image Processing)

MSc THESIS

2013-2014

DERECK D WEBSTER

Automatic Rain Drop Detection for Improved Sensing in Automotive Computer Vision Applications

Supervisors: Dr Toby Breckon / Dr Mark Lee Stillwell

04 April 2014

This thesis is submitted for the award of MSc by Research

ABSTRACT

The presence of raindrop induced distortion can have a significant negative impact on computer vision applications. Here we address the problem of visual raindrop distortion in standard colour video imagery for use in non-static, automotive computer vision applications where the scene can be observed to be changing over subsequent consecutive frames. We utilise current state of the art research conducted into the investigation of salience mapping as means of initial detection of potential raindrop candidates. We further expand on this prior state of the art work to construct a combined feature rich descriptor of shape information (Hu moments), isolation of raindrops pixel information from context, and texture (saliency derived) within an improved visual bag of words verification framework. Support Vector Machine and Random Forest classification were utilised for verification of potential candidates, and the effects of increasing discrete cluster centre counts on detection rates were studied.

This novel approach of utilising extended shape information, isolation of context, and texture, along with increasing cluster counts, achieves a notable 13% increase in precision (92%) and 10% increase in recall (86%) against prior state of the art. False positive rates were also observed to decrease with a minimal false positive rate of 14% observed.

ACKNOWLEDGEMENTS

I wish to thank Dr Toby Breckon for his time and commitment during my project, and for the help given in patiently explaining things for me. I also wish to thank Dr Mark Stillwell for his never tiring commitment to proofreading and getting up to speed with my project. Without them, this thesis would never have come together in the way it did.

I wish to thank my partner, Dr Victoria Gortowski for allowing me to go back to university, supporting me and having faith that I could do it, without which, I do not think I would have. And last but not least, Lara and Mitsy. Thank you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

1 INTRODUCTION

The work in this thesis is to expand on prior state of the art research into the investigation into raindrop detection and verification. We research the topics of how raindrops appear to the camera, the various methods of detection and investigate various methods of improving the verification of potential raindrop candidates after detection. We particularly investigate the use of Hu shape moments as an extended shape descriptor (Section 4.4), and raindrop context isolation (Section 4.3), where the raindrop pixel information is removed from the background, with the aim of reducing the high false positive rates seen when using the techniques described in [1].

1.1 Motivation

Computer vision techniques are becoming increasingly common in vehicles, and where Driver Assist Systems (DAS) were once only in luxury cars, they are becoming more common in mid-range and budget cars. Indeed, the functionality of these systems is growing as they become increasingly mainstream with functions such as speed sign detection [2] as depicted in Figure 1 where the camera detects the speed sign and notifies the driver, road feature detection [3] headlight detection [4] and road marking recognition [5] with more advanced features appearing that rely on computer vision.

Figure 1 - Typical example of 'Speed sign recognition functionality. Source: carsafetyrules.com (2014), http://www.carsafetyrules.com/has-volvo-created-the-ultimate-small-family-car/1106/

However, as driver reliance on these systems will doubtlessly increase, both for safety, i.e. auto stop safety systems as shown in Figure 2 that bring the car to a stop after detecting a possible collision threat, and autonomy, where the car drives itself with no user input, the information gathered by the on-board camera(s) has to have a high level of accuracy. This is to enable the decisions made by the DAS system or driver to be made with information that is deemed correct for its given situation.

Figure 2 - Example of auto stop safety system. Source: Euroncap.com (2014), http://www.euroncap.com/

Background investigation conducted before commencing this work, however, showed that the effects of raindrop distortion on the speed sign detection code in [2] caused a 50% decrease in detection rates in rainy scenes compared to clear scenes.

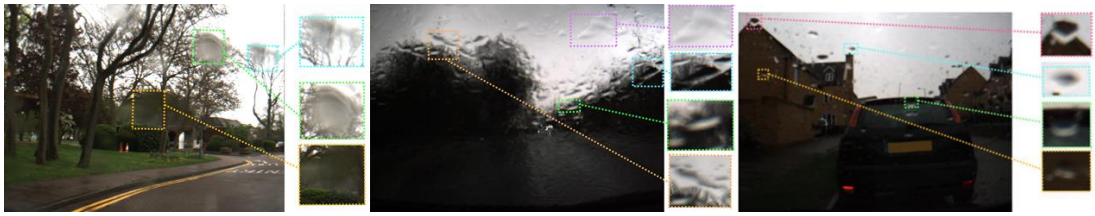
Figure 3 - Speed sign detection

As seen in Figure 3, although the scene remains the same between the two images, the addition of raindrops causes the failure of the algorithm to successfully detect the speed sign (as indicated by the white border around the sign). Although this may only impact one frame in this instance, over a period of time the cumulative effect of these misses may result in a DAS system that requires a succession of successful identifications taking longer to return a result. The effect of this can manifest itself in delays to systems reliant on the output from this system, or information being presented to the driver that is pertinent to the current situation.

Adverse weather conditions have also been observed to have a negative impact on other common place visual surveillance tasks [7,8]. Despite this however, the investigation into the potential impact of raindrop distortion upon automotive visual techniques [1, 2, 3, 4] remains in its infancy [1,9].

1.2 The Raindrop

During initial data gathering it became apparent that raindrops have different characteristics. This enabled us to separate the raindrops into three distinct groups. Non-Defined raindrops have no discernible edges and are barely visible to the camera, defined raindrops have sharp defined edges and very apparent in the scene and rain streaks which occlude large regions of the image and have no defined edges.



A B C

Figure 4 – Typical raindrop shape and colour variation examples within an automotive sensing application

Figure 4 highlights these raindrop classifications with three examples of wet weather scenes showing various raindrop shape and colour variations within an automotive setting. Figure 4A shows non-defined raindrops that have no discernible edges. These are barely visible but still occlude the region that they lie over, causing significant localised blurring of the background. Such raindrops are hard to detect due to the lack of difference in brightness compared to the surrounding area and lack of defined edges. Figure 4B shows heavy rain whilst travelling at speed. Raindrops in this circumstance tend to join and cause rain-streaks, which travel up the windscreen as the vehicle is moving forward. These regions cause significant blurring of the background, covering large regions, and

occluding entire objects in some instances. Raindrops here are difficult to detect as because of the lack of contrast at the edges. The rain streaks also occlude such vast regions of the image, they themselves become background. The raindrops in Figure 4C were observed when a vehicle is at rest in light to moderate rain. The raindrops in this instance can be seen to follow the model described in the work of [9] and [8], where the raindrop has a dark region at the top and a light region at the bottom and can loosely be described as round. However these raindrops are seen only on rare occasions, typically when the vehicle is at rest (e.g. at traffic lights), which makes the application of such models less appealing for non-static automotive applications.

It is therefore clear that it is not possible to effectively detect raindrops in a variety of conditions by means of a model as seen in [9] or using a change in pixel / boundary intensity as investigated in [8].

1.3 Aim and Objectives

The aim of this work is to review and improve the prior independent work reported in [1] with an explicit focus to improve verification rates and reduce the high false positive detection rates for 87% found during our initial investigation.

This aim can be achieved through the following key objectives:-

1. Evaluate the current 'state of the art' saliency method for raindrop detection [1]
2. Investigate differing techniques to lower false positive and false negative detection rates
3. Evaluate accuracy of detection rate, both in varying rainy weather conditions and when no raindrops are present
4. Compare and evaluate achievable results between current state of the art [1] and a range of our novel adaptations proposed (Use of shape features, raindrop context isolation, varying cluster counts, use of alternative classifiers [10])

The work of this thesis can be broken down into two key stages. The first stage is to derive a saliency map from the incoming video images following the approach used in [1]. This results regions of interest (regions that contain potential raindrops) which will form the initial input into our verification stage. Within the second stage, we consider the use of an extended feature description model, which combines both the additional use of shape descriptors [11] and raindrop pixel isolation from the scene context. Verification is then carried out using a data driven classification approach following a bag of visual words model, leveraging the power of both support vector machine (SVM) and random forests (RF) classifiers, over a range of

varying cluster counts. We then provide comparative results for each machine and technique, providing a comparison of accuracy in various rainy and non-rainy conditions.

2 LITERATURE REVIEW

Although it is understood as a potential issue for any computer vision application that is exposed to the elements, the research that has been conducted on how to detect and remove raindrops is rather sparse [1,9]. Of the research that has been conducted, the focus is concentrated on looking at raindrops that are moving through a scene [8], as in rainfall in camera footage [12] or on adherent raindrops on images where the background remains static, e.g. CCTV in low traffic areas [13]. Research into how to detect and remove raindrops in vehicle and how it affects Driver Assist Systems (DAS), however, is a relatively novel field.

In this literature review we cover the research conducted so far regarding the detection and removal of raindrops [8,7] and a look at how a camera sees rain [14]. Both stationary cameras, where the camera is fixed and the background does not change [12] and non-stationary cameras, where the camera traverses through the scene [16,17] are considered. Models[17] and visual data [1] approaches are also discussed to see if these approaches may be extended to in-vehicle use. We further investigate the methods that are considered for raindrop verification (where a region of the image, or patch, is analysed to see if it has a raindrop), and the methods used if the raindrop is being removed.

2.1 Raindrop Formation

The understanding of when a camera 'sees' or detects raindrops is fundamental to the further understanding of how and when a system may be used to detect raindrops. It was found in [14] that rain produced spatio-temporal intensity fluctuations which caused temporary increases in pixel intensity in a specific region. The visibility of the drop is dependent on the camera settings and altering the exposure time, the depth of field, or both, reduces the appearance of raindrops within the scene. As the exposure time is increased, the raindrop becomes increasingly blurred so that that it is not visible on the final image. However, an increase of exposure time may result in motion blur from any movement in the scene and is unsuitable for an application where the camera is not stationary or the scene contains any movement. By changing depth of field the foreground is sufficiently thrown out of focus so as to render raindrops invisible to the camera. However, in order to change the depth of field, the focus has to be known. Another drawback is that limited scene information is obtained when the depth of field is small, making it difficult to track objects through a large field of view.

Subsequently, research has investigated the spatio-temporal intensity fluctuations that raindrops cause and in [8] the authors derived a model based on the photometric and dynamic properties that are seen as a drop flows through a scene. This provides the ability to detect the drop in the initial instance and then estimate its trajectory. The assumption is made that all drops are the same size, are spherical, are uniformly distributed throughout the scene, and are falling at a constant velocity. Therefore the dynamic properties can be used to estimate how the raindrop will flow through the scene frame by frame. For example, if a drop was detected in the far top left of the frame, and in the next it was seen to move several pixels to the right and down, an estimate can be made on how the raindrop is going to continue through the scene, top left to bottom right. The work also found that as a raindrop moves through a scene it generates an intensity change at pixel level. Where pixels were found to have high intensity values, it can be assumed raindrops are present, and vice versa.

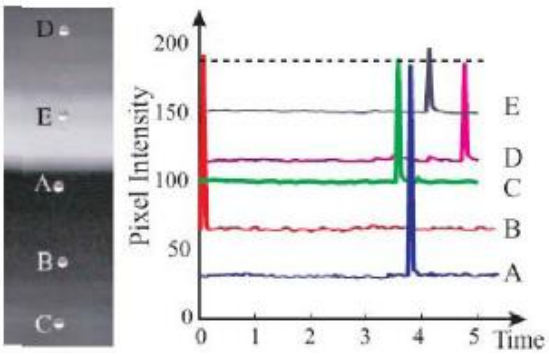


Figure 5 – Photometric Properties. The pixel intensity changes as the raindrop fall through the scene.[8]

As seen in Figure 5, whilst a raindrop moves through a scene, the photometric properties of the raindrop vary the intensity of the pixel and with the absence of motion blur, the intensity of the drops is almost the same and is independent of the background. Using the photometric properties, raindrop candidates on the n^{th} frame can be ascertained as bright regions. A binary field can be created on the $n^{th}+1$ frame using the dynamic properties and compared with the n^{th} frame to get a correlation magnitude where bright regions have raindrops present. These methods can be used to remove false positives; when a region is misclassified as a raindrop, the recovery value for the pixel is the average or *n-1* and *n+1*. However, experiments have found that this system struggles with bright backgrounds and complex scenes as there is very little intensity change between the scene and the raindrop, therefore rendering this approach inoperative in sunny showers. This method is also cannot to be used for non-stationary scenes, such as in-vehicle

applications or heavy rain as this throws the estimated trajectory. The assumption that all drops are the same size is also false. Figure 4 shows that raindrops differ in size and intensity depending on various conditions.

2.2 Raindrop detection and removal – stationary case

A noise reduction method for removing drops or other noise objects has been trialled where a succession of images are captured over a period of time with noise objects moving over a stationary background [12]. The theory is that as a number of images are taken over time, all parts of the scene at some point will have been visible and not be distorted or occluded in anyway. Therefore, to reconstruct the image, all the individual images are initially added together and a grey level histogram is calculated then averaged to create a 'combined histogram'. A grey level histogram is then calculated for each, individual frame which is subtracted from the combined histogram previously calculated, thus giving a reconstructed image. However, this method cannot be performed in real time (as the initial combined histogram requires all recorded frames), relies on the fact that noise is propagating throughout the scene and therefore all regions of the image is seen at least at one point in time, and the background remains static as in the case of a photograph or fixed position CCTV camera with little to no observed movement.

By analysing each pixel in a frame and looking for intensity changes at that pixel site with regards to background brightness, it is possible to measure the photometric properties of rain. In [8], an algorithm to derive an estimation of intensity of a given frame was created using a Kalman Filter [18], this deals with initial background brightness. The assumption is made that the camera and captured object are fixed, and therefore a pixel point is the same in subsequent frames. The estimated intensity is calculated for frame *n+1.* Raindrop candidates are identified via analysis of each pixel in collation with the generated photometric properties with the estimated intensity used to remove the raindrop. Although the Kalman Filter can be used in real-time, it has the same limitations as the method used in [8], where the assumption is made that the camera and background are not moving, or indeed, are moving very slowly as to be considered stationary for a number of frames, therefore the scene does not differ in subsequent frames. This, therefore, has limitations on automotive use.

The use of optical flow, defined as an estimation of the motion of the camera, has been approached as a solution to detecting raindrops [19]. Optical flow is used to estimate the motion field where sharp changes in intensity are viewed, and therefore is capable of estimating motion trajectories of objects. As the movement of non-raindrops is known for the frame *n+1,* subtracting this frame from *n*, gives us regions where raindrops are not present. Using this knowledge and the photometric

properties leaves regions where raindrop movement is likely. The rain is then removed by updating each pixel in the rain region with the weighted sum of its spatial-temporal neighbourhood. However, this requires a number of frames to be gathered, that there are no great changes in the background between frames, and that the raindrops are the only source of fast movement, as in transiting through the scene. As a further drawback, if rain falls at too high of velocity, the estimation of its trajectory is likely to be inaccurate.

2.3 Detection and Removal – non-stationary case

Whilst section 2.2 discusses approaches that have been taken to remove raindrops that are falling through the scene, or are occluding a region on a static image; it has not been discussed where raindrops that are adhered to a lens or protective screen as shown in Figure 4 and thereby occluding a region of the image. It has neither been discussed where the camera is moving or a scene is moving around the camera.

One approach in non-stationary cases is to find potential raindrop candidates or noise by estimating camera motion from spatio-temporal information across multiple frames [13]. As shown in Figure 6, this approach uses multiple frames as a camera either passes through or pans across a scene, utilising optical flow (where a visual scene change is caused by the relative motion between the observer and the scene) to estimate camera motion. Once the images are stacked, taking a cross section will result in the trajectory of any background object to be straight and any adherent noise to appear curved. This final, stacked image is then subtracted from the initial image (Figure 6a) removing all background objects and leaving the curved adherent noise line (Figure 6b) and therefore providing the region that contains the adherent noise (Figure 6c).

If the camera motion through the scene is known, we do not need to use any estimation and can use projective transformation to predict where the next image sits on the image plane [12]. Again, stacking the images in chronological order (Figure 6a) and taking a cross section results in trajectory information of both the background objects and adherent noises (Figure 6b). However, objects that are moving throughout the scene could potentially appear as fixed points and be incorrectly identified as raindrops, making this method unsuitable where high traffic (i.e. high traffic roads) or footfall (high pedestrian areas) could be moving throughout the scene.

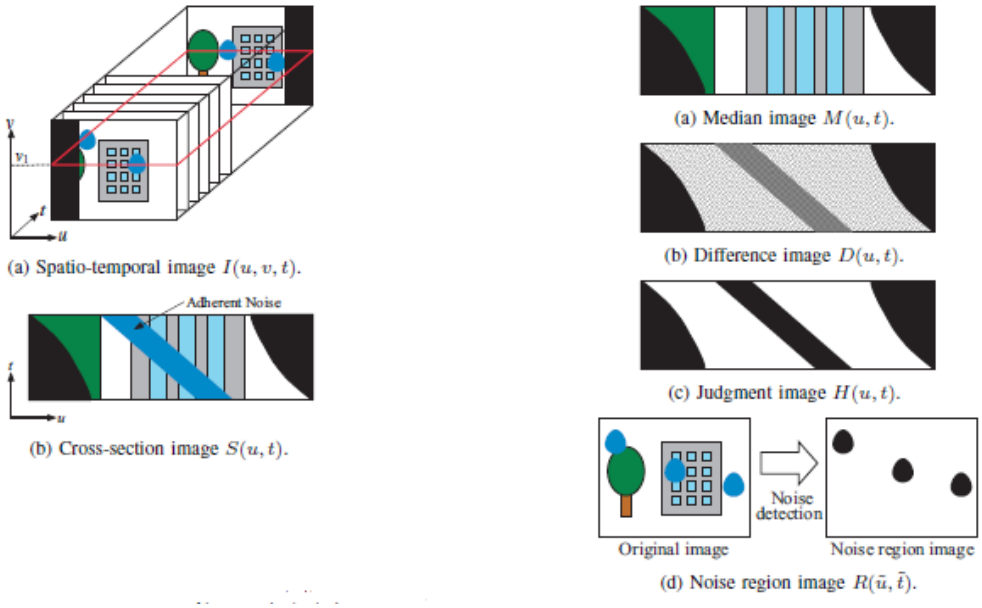


Figure 6 - Example of chronological stacking and extraction of trajectory information. As the camera is rotated at a constant velocity a succession of frames are built up. Clipping a cross section and applying a projective transformation, the trajectories of the background will become vertical straight lines (grey, green and light blue), and adherent noises will become curves (dark blue). [13]

The noise regions that are detected in the above approach are removed using an image restoration technique first described in [20]. Images are first deconstructed into structure images and texture images allowing inpainting and texture synthesis to take place. Texture is synthesised by first taking a template next to the identified region requiring in-painting and looking for a best match to this region based on colour, contrast and estimated texture behind the occluded region. The candidate pixel to be replaced is then selected beside the pixel that best matches the occluded region and after in-painting, the best match texture found is added. The technique here is fairly robust, showing good results with adherent objects such as mud and raindrops that are clearly defined as shown in Figure 4C. However, the method does heavily rely on raindrops being visible to the camera, and therefore will struggle with raindrops that fall into the other two categories (Figure 4 A and B)

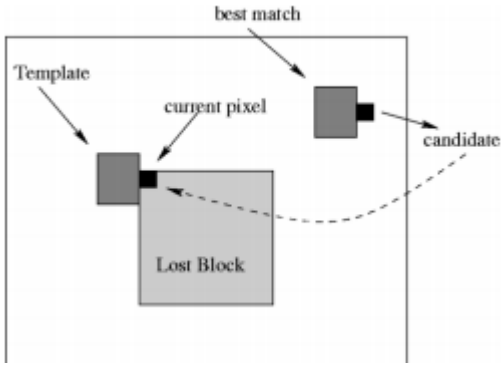


Figure 7 – Example of texture synthesis. The template is generated and a best match is searched for across the image. The candidate is then used to generate the texture that is missing from the incomplete region. [20]

[15] uses panning or tilting a camera in a similar fashion to [12] and [13], however instead of capturing multiple images over the entire pan of the camera, two images are taken to produce a 'pseudo stereo' image . An initial image is taken as a reference image and the camera direction is then rotated along a fixed, known axis, and a second image is taken. This second image now includes visible information that was occluded in the previous image. This second image is transformed with projective transformation and after thresholding to isolate brighter objects further, compared with the first initial image

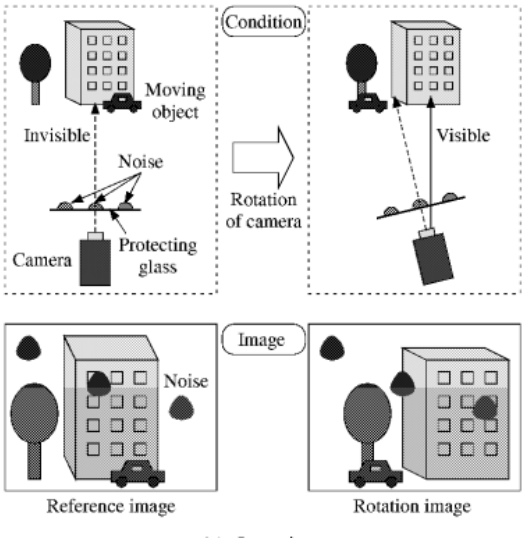


Figure 8 – As the camera is rotated along a fixed axis, the items that were occluded are now visible. Knowing this rotation of the camera it is possible to work out if adherent raindrops are in the same position in both images. [15]

By knowing the distance the camera direction was altered, and because the raindrop is adhered to the lens, it is possible to calculate if the drop is present in the same place in both images. After calculating the disparity between the two images, images are then reconstructed by using pixel information from corresponding non-

occluded regions on the second image. However, before noise or raindrops can be removed from the images, the camera must have moved and the rotation distance be known. The scene must also remain relatively static; otherwise the algorithm could potentially remove objects that have moved whilst the camera had been rotated, especially objects that move at the same rate as the panning movement of the camera. This limits the effectiveness of this approach for in-vehicle use, where the scene is very dynamic. The same assumption made in [12] and [13] that raindrops will follow the type shown in Figure 4C is also made here.

In [21] and [22] the need for panning or tilting the camera in removed by utilising a stereo setup [21] or by use of multiple cameras [22]. A disparity image is created by template matching of the two stereo images and correlations are thresholded to discard unreliable points. The points are also investigated to ascertain if they correspond one-on-one. Where the disparity value is high, an object is close to the lens of the camera, and therefore potentially a raindrop or other adherent noise. Raindrops are then removed in a similar fashion used in [15]. This has the benefit of being usable with unknown movements, such as manually controlled CCTV cameras, where the angle of change is unknown, but in order to work, the background has to be relatively still to limit any disparity to raindrops and that the raindrops are visible to the camera.

The approaches above have also been demonstrated to work where adherent raindrops are static, but the camera is transecting through a scene or the background is not static [23]. It is surmised that the temporal change of raindrop pixels are smaller than that of non-raindrop pixels on moving backgrounds, i.e. bright regions where raindrops exist are bright over multiple frames. The approach to remove raindrops is to blend affected regions and use a video completion technique [24], where missing data is assumed to reappear somewhere in the frame, although this approach is used only for regions that cannot be restored.

2.4 Detection – Model and data driven approach

In [25] work has been conducted to understand the geometric- photometric properties of a raindrop and image correlation so that an accurate model could be constructed. Dubbed the 'Raindrop *Intelligent Geometric Scanner and Environment Constructor*' or *RIGSEC*, the work concentrates on the analysis of how light refractions through a raindrop so that a model can be generated. A specific region on an image is then distorted using this raindrop refraction model, thus giving the region an appearance as if a raindrop is present. The assumption here, however, is that raindrops follow one specific model.

Further work was undertaken to place these raindrops on video footage in regions that contain potential raindrop candidates [17]. Images were blurred using an adaptive bandpass filtering method based on computing the Difference of Gaussians (DoG) for each pixel. Once this had been computed, the difference between the regions containing raindrops in the original image and regions containing raindrops in the blurred image are greater where raindrops do not exist. This is due to the raindrops blurring the region that they occlude. Verification was performed by placing a model of the raindrop in likely regions and measuring the correlation in their intensity values.

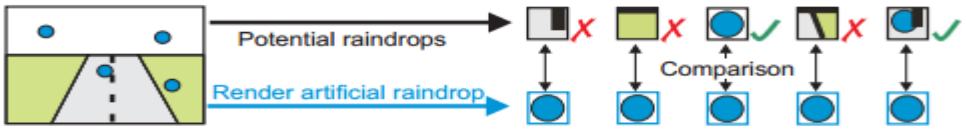


Figure 9 – A flow chart of raindrop detection with RIGSEC. Raindrops patterns are compared to the artificial drops using intensity based correlations. [25]

Figure 9 shows how the model raindrops are compared to regions of interest that are selected by using Surf feature detection [26]. The artificial raindrop is compared to the selected region of interest based on intensity correlations. If these match, a raindrop is present. The downfall of this method however is that the generated model may not match the raindrop present in the scene.

A theory of whether or not raindrops can be described as salient features has been investigated using methods to extract colour, texture and shape of raindrops with varying results [1]. The idea is that raindrops are high contrast regions, which are usually texture-less and round, compared to the surrounding regions. Utilising this method means that the camera does not need to be traversing through the scene, nor do multiple images have to be captured before detection and removal techniques can be applied. Images were processed following the flow in Figure 10 gathering colour and texture feature maps with Hough circle transform [26] being used to find round items. These were then combined to find the brightest regions in the scene.

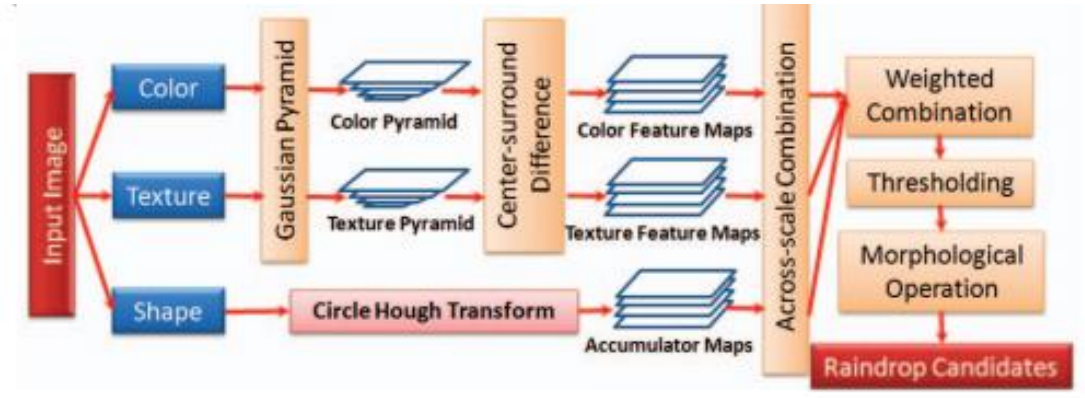


Figure 10 – Generation of Saliency map. Images are passed through a Gaussian pyramid and then subtracted from each other across-scale to work out the texture and feature maps. Hough transforms are used to find regions that are circular within the image. [1]

The images were combined across scale by means of a weighed combination of the colour, texture and Hough circle feature maps. Using a weighed combination means that certain features maps have more influence that others in this instance. The resulting images were then thresholded to create a mask which was then eroded and dilated to remove any noise. This mask is used to isolate the regions of interest, or potential raindrop candidates, by copying the information from these regions only. Doing this means less data has to be analysed, thereby meaning a reduction in required computational resources. These regions are then passed to a GMM [27] (Gaussian mixture model) clustering method, and studied by an SVM (support vector machine) to ascertain if the selected region contains a raindrop. Results from this method are fairly strong showing a 79% true positive detection rate. However, false positives are not reported.

It is unclear; however, whether describing raindrops as salient, or objects that are distracting is accurate. As saliency is a complex subject, it is therefore is discussed further in Section 3. It is also noted that rain may be difficult to detect in different contrast scenes, where there is little to no contrast difference between raindrops and background scenery. Thereby, this method could be susceptible to high false negative rates in these described conditions.

2.5 Raindrop Detection – optical approach

Research has been conducted with regards to using existing set ups of DAS systems for automatic triggering of windscreen wipers, but with the addition of another lens, mirror and LEDs [28].

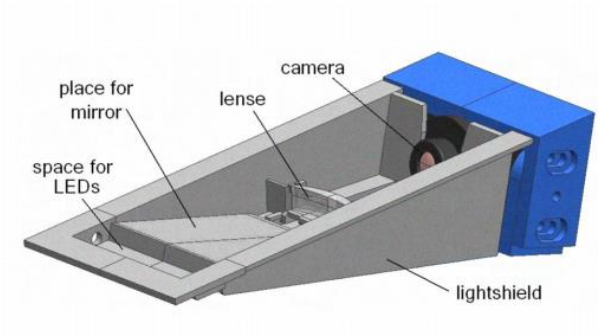


Figure 11 – Modification of an existing DAS system for adjusting focus to include objects adhered on the windscreen. [28]

Figure 11 shows how this modification to an existing DAS system could be implemented to bring adherent objects on the windscreen into focus. LEDs are used to illuminate the DAS field of view for night driving with a baffle in place to stop light leakage distracting the driver. The purpose of the additional lens is used to bring objects that are usually close to the lens, such as raindrops, into focus and to throw the background out of focus only in the lower region of the cameras field of view. LEDs were used so that the system can be used at night. Due to the focus being strongest on the windscreen surface, all background objects were blurred, creating only slight shadows, therefore allowing the use of the Sobel operator [29] for image evaluation. Edge pixels were then accumulated column wise to create a 1-D distribution graph, or 'rain index'. Once a set threshold had been met for the pixel count, a rainy frame counter increased and once 10 consecutive frames had been monitored, the wiper was triggered.

2.6 Summary

Although several studies have been conducted into raindrop detection and removal, only two are explicitly aimed to work within driver assist systems, the model approach [28] following the photometric properties of raindrops and [1], a feature based detection technique. The work of [1] does not require any modification to existing DAS camera setups and is more aligned to the research we wish to undertake. It is also stated that it is able to detect raindrops without using a temporal approach and can be described as 'real-time'. Therefore the work in this thesis extends the work of [1] specifically targeting second stage raindrop verification, by using an extended descriptor model based on Hu Moments and Raindrop Context Isolation.

3 RAINDROP CANDIDATE IDENTIFICATION

Within this section, we first investigate in section 3.1 the saliency background and how it works in biology, specifically with mammals, and how this ties in with how raindrops can be described as salient items. We investigate in section 3.2 how we can mimic this process saliency process computationally and then move on to consider the proposed modified saliency model (Figure 15) provide in [1] for raindrop detection in section 3.3. This section considers how the raindrops discussed in section 1.2 can be described as salient , and moving on to how we implement this modified saliency map for raindrop detection.

3.1 Saliency in Mammals

The work of [1] stands on the assumption that raindrops can be identified as being salient, but to understand this or see if this is an accurate assumption, we need to understand saliency and what it is.

Saliency can be described as a "striking point or feature" [30] which lies in the scene. In this case, the term "feature" may be either an object, as in a region with clear boundaries, or some other part of the scene that may not be associated with any singular object, such as a colour or texture. This salient feature is usually more prominent in the scene than any other visual distractions [31], and thereby easily stimulating the parietal cortex [32] within mammal brains. It should be noted that the saliency of an item is dependent on its surroundings and situation; what may be salient in one surrounding or situation, may not be in a different surrounding and situation. For example, it is commonly believed that saliency improves the ability to focus on targets that are of interest, e.g. during hunting or foraging in primates [32]. In this case, the prominent feature would be prey or fruit respectively, however during mating season; potential mates will become more salient than other, usually important, objects (e.g. food).

Two different models have been derived for how the brain sees, or points out to us, salient objects. Taking the hunting example, this can be described as *top down* or *user driven* [31, 32], finding the object of desire is more important than the examining all individual objects within the scene.

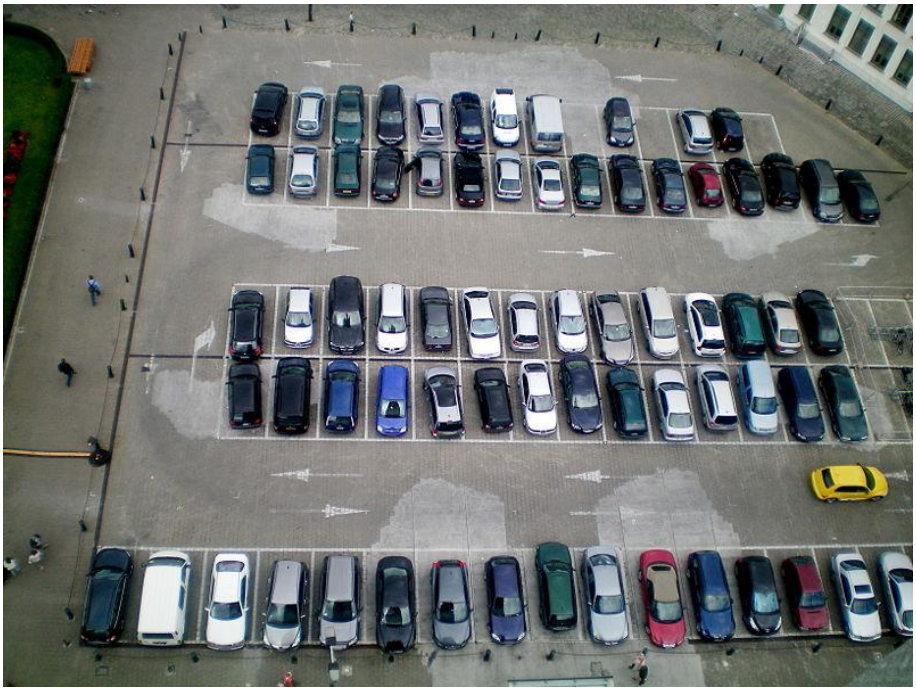


Figure 12 - Image of car park from above. Source: http://en.wikipedia.org/wiki/Parking_lot (2014)

Figure 12 shows a car park when viewed from above. If you are looking for a blue hatchback car in a busy car park, all other blue objects such as a car or a person wearing a blue coat, will become salient and stand out. This is driven by the desire to find a blue object. Closer (post saliency detection) inspection is required however, to ascertain which of the objects are cars, and which ones are not. Therefore, colour, not the object, can be considered to be important attribute, in this scenario.

Considering Figure 13 with the top down approach, if we were to try and pick out the purple brick that brick will stand out immediately due to the colour and the brick size or shape will not be considered. Alternatively, if we were to look for the four-across brick instead (within Figure 13), the red brick would stand out and the others would not be considered, due to looking for a specific shape. Finding the object of desire (either in shape or size of the object or in colour of the object) in this approach overrules examining other objects that are present in the scene.

Figure 13 – Top down approach applied to Lego bricks

The opposite of the top down approach, is the *bottom up*, s*timuli driven* or *saliency based visual attention*. This approach describes how the brain spots 'the odd one out' or an unusual features within a scene that might be worthy of further investigation. If we consider the image in Figure 13 again, but without a specific object in mind, the red brick may stand out, or the brain may notice the purple brick instead. This is because these bricks are different from the others, with the purple one standing higher than the rest, or the red one having more spots on it. The colour attribute does not have an influence in this circumstance. Of course, it should be noted that colour can be an attributed feature in the bottom up approach too. Within Figure 12, without an object of desire to find (as in the *top down* approach); the yellow car may attract your attention. This is because it it's the only car of colour yellow within the scene, although, the direction that the car is facing may also be contributing to the saliency of this particular object.

Objects such as these can be described as distracting and can overrule top down saliency[34]. As raindrops on a car windscreen can become a distraction, the process of detecting them, therefore, falls into the bottom up approach of saliency.

3.2 Saliency in Computer Vision

Computationally, we can mimic this saliency process, isolating one single location of potential interest by generating a saliency map, a map of the process to detect regions of interest. Doing so gives us the ability to describe to a computer visual system important items in the current frame. The work of [35] provides a model of how a scene can be analysed by a computer vision system to provide an output, a saliency map, of what objects within the scene demand the most attention.

Figure 14 shows the generalised model of the program flow for determining the saliency map. After working out the filters of the colour, intensity and orientation, it becomes apparent that not just the colour of an object but also the intensity and the orientation is important. Once the maps are combined across all scales and across all three filters, the saliency map is generated. A 'winner takes all' approach is taken, i.e. the area with the highest pixel count of 255 is the region where interest should be focused.

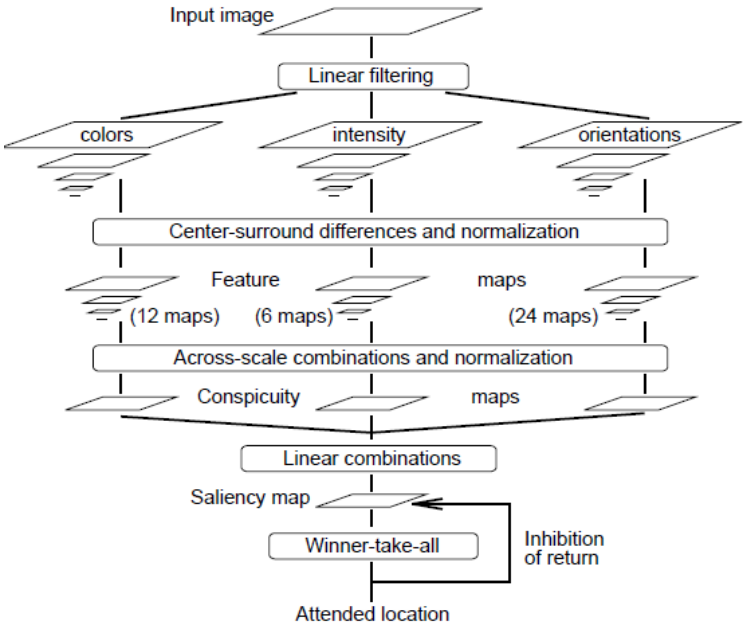


Figure 14 – The generalised model of the program flow to produce the saliency map. [35]

Using this we can predict how the gaze of a person may proceed to take in a particular scene. The work of [36] has looked into this, attempting to predict the tracking of a person's gaze with faces in a scene. The understanding of how a person's gaze may move about the scene may help further research into the positioning of road markings [3] and speed signs [2].

3.3 Saliency Map Generation for Raindrop Detection

Considering Figure 4, one attribute that makes raindrops salient is that they have a higher contrast than the surrounding regions. The work of [1] utilises this fact to generate a modified saliency map (Figure 15) using colour and texture and Hough circle transforms [29] being used for shape description to extract raindrop candidates (regions that contain potential raindrops). Within this work, we further extended this approach by using an extended descriptor model based on Hu Moments [11] for extended shape information. The saliency map is generated in five steps, after separation of the colour space; intensity, texture and shape

features maps are generated and then weighed summed to provide the final saliency map of the scene. These are discussed further in the following sections.

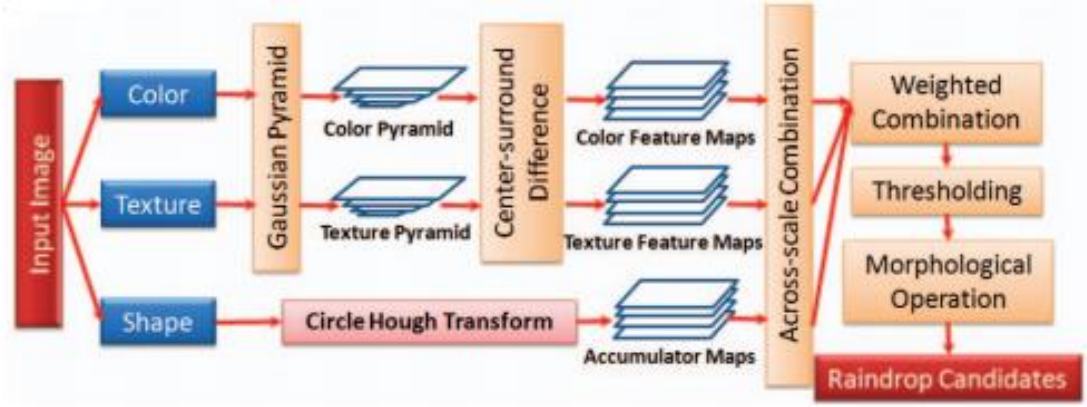


Figure 15 – Saliency Map. Colour, texture and shape are used to generate potential raindrop candidate regions. [1]

3.3.1 Colour Space Separation

Before generation of our feature maps, the maps of our salient features, the image is separated into the XYZ colour space as the work conducted in [37] has shown that this results in an improved raindrop discrimination between them and the foreground.

An input image for a typical road scene with raindrops present (Figure 16A) is firstly separated into the XYZ colour space (Figure 16 B – D) where Figure 16B, Figure 16C and Figure 16D equals the X Y and Z colour channels respectively. The resultant colour space images (Figure 16 B-D) are then passed through a multi-scale Gaussian pyramid [26] (n = {0 to 4]}) with the n^{th} corresponding to subsampling by a factor of 2^n as a means to identify features at different image scales. These images are then used to generate our intensity feature maps and our texture feature maps.

A

B C D

Figure 16 – Exemplar images in XYZ colour from original RGB image (A).

3.3.2 Intensity Feature Maps

Intensity feature maps, *T*, are created for each XYZ channel by detecting intensity differences between the centre pixel and its neighbours, thus giving us the intensity contrast features *I*. The centre pixel is a pixel at scale $c \in \{0,1,2\}$, and the surrounding corresponding pixels are at scales $s = c + d$ with $d \in \{2,3,4\}$, $s \leq 4$. For each of the XYZ colour channels we compute these differences as a set of maps for six different pairs of *(c,s)* values, *(0,2), (0,3), (0,4), (1,3), (1,4)*, which satisfy $c \in \{0,1,2\}$ and $s = c + d$, $d \in \{2,3,4\}$, $s \leq 4$: $I(c,s) = |I(c) \updownarrow I(s)|$ (where $\updownarrow$ denotes across scale differences between the two maps). Furthermore we obtain a secondary saliency map from the intensity features by using the same cross-scale sum from the six feature maps obtained above, over all three XYZ colour channels, *i*, as follows (where (+) denotes cross-scale addition):-

$$\hat{\bar{I}} = \sum_{i=0}^{2} (+)\ ^{2}c = 0\ (+)\ ^{4}s = c + 2\ I_i(c,s)$$

Equation 1 – Obtaining intensity feature descriptors. (+) denotes across scale addition

Figure 17 gives examples of the images that the intensity feature detector produces across various differing weather and lighting conditions. Images across the top are the original images, with images across the bottom being feature rich intensity feature maps.

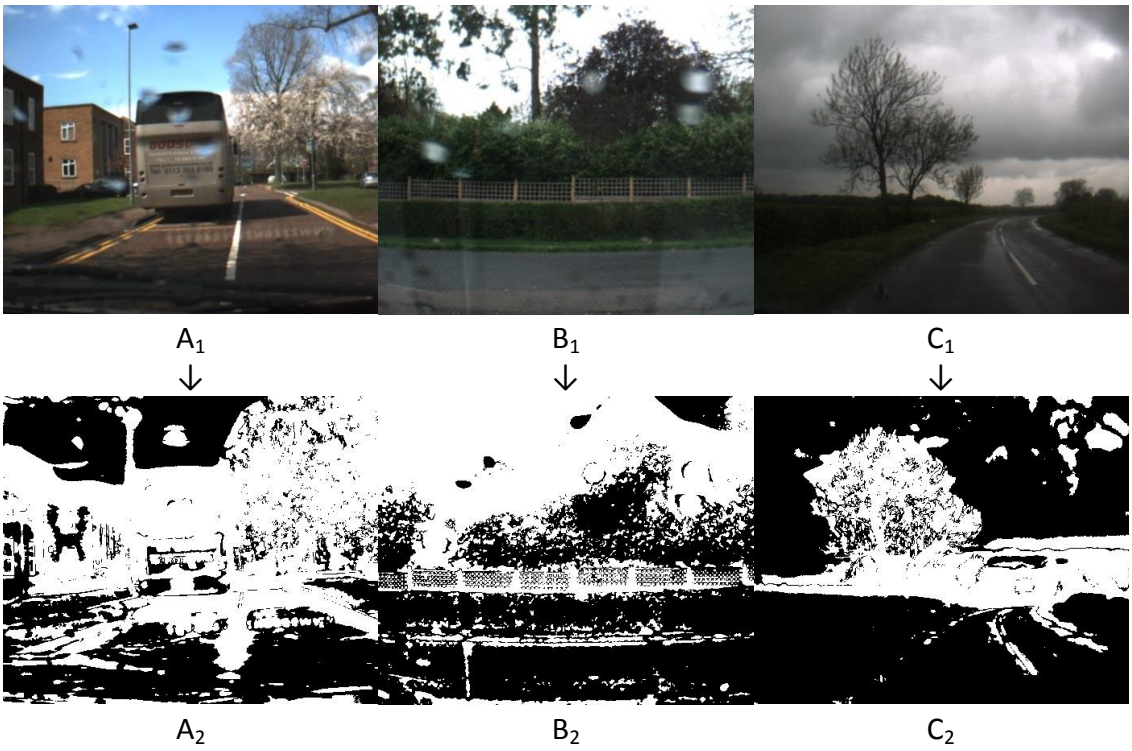


Figure 17 - Exemplar images after intensity feature detection with their respective original image

In Figure 17 A_1, we can see that some of raindrops differ in intensity sufficiently from the background with the result that they are identified by the feature detector reasonably well (Figure 17 A_2). However, a few are missed as the intensity change is not great enough for the feature detector to identify them. The intensity of the raindrops in Figure 17 B_1 varies greatly from the background and therefore most, if not all, are identified by the feature detector (Figure 17 B_2). Figure 17 C_2 shows that some areas are highlighted due to a high intensity change to the surrounding regions. However these regions do not contain any raindrops, with the dark clouds and bright sky causing this incorrectly identified regions.

3.3.3 Texture Feature Maps

Subsequently, separate texture feature maps, *T,* are created following a similar methodology as outlined above for the intensity feature maps, where the cross-scale sum is again taken from all six feature map pairs, *(c,s),* across all three XYZ colour channels, *i,* as follows (where (+) denotes cross-scale addition),

$$\tilde{\tilde{T}} = \sum_{i=0}^{2} (+)\ {}^{2}c = 0\ (+)\ {}^{4}s = c + 2\ T_i(c, s)$$

Equation 2 – Obtaining texture feature descriptors. (+) denotes across scale addition

However the calculation for these maps differ from the calculation of the intensity feature maps (section 3.3.2) in that we convolve with the second order edge detector, the Laplacian of Gaussian [29] with the kernel size, *K*, set to three. This results in the exemplar images shown in Figure 18. The top images are the original image and the bottom images are the generated texture feature maps.

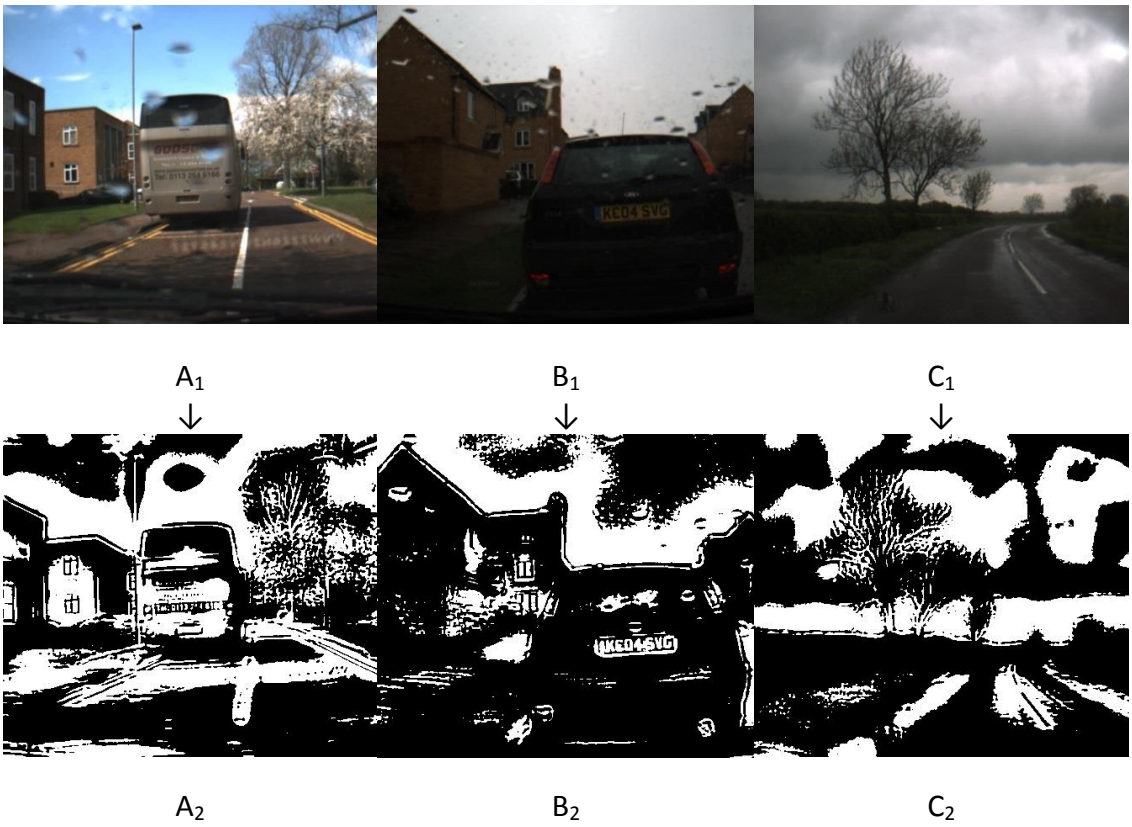


Figure 18 – Exemplar images after texture feature detection with their respective original image

In Figure 18 A_2, although the raindrops are apparent in the original image (Figure 18A_1), they lack defined edges, therefore the edge detector fails to identify them. Figure 18 B_2 shows the example of raindrops that are clearly defined and have sharp, contrasting edges to the background (Figure 18 B_1). The edges are apparent enough for the feature detector to identify drops that are not immediately apparent in the original image (Figure 18 B_1). Figure 18 C_1 gives an example of where no raindrops are present for the detector to identify. However edges are still detected such as the edges in the clouds and the edges of tree branches.

3.3.4 Hough Circle Transform

Further shape information, *S*, is added on a per pixel basis by detecting circular raindrops using Hough circle transform [38]. The Hough circle transform is derived from the Hough transform for lines, but, uses the parametric form for a circle as described in Equation 3.

$$x = x_0 + r\cos\theta$$

$$y = y_0 + r\sin\theta$$

Equation 3 - Hough circle transform

Here x_0 and y_0 represent the coordinates for the centre of the circle, with r being the radius of the circle.

Circular objects are identified using the original images scale due to the potential for raindrops to disappear after sub-sampling in the Gaussian pyramid phase. This is done across all three colour channels following [1] (pixel location $S_{(i,j)}$ = 1 for detected circle).

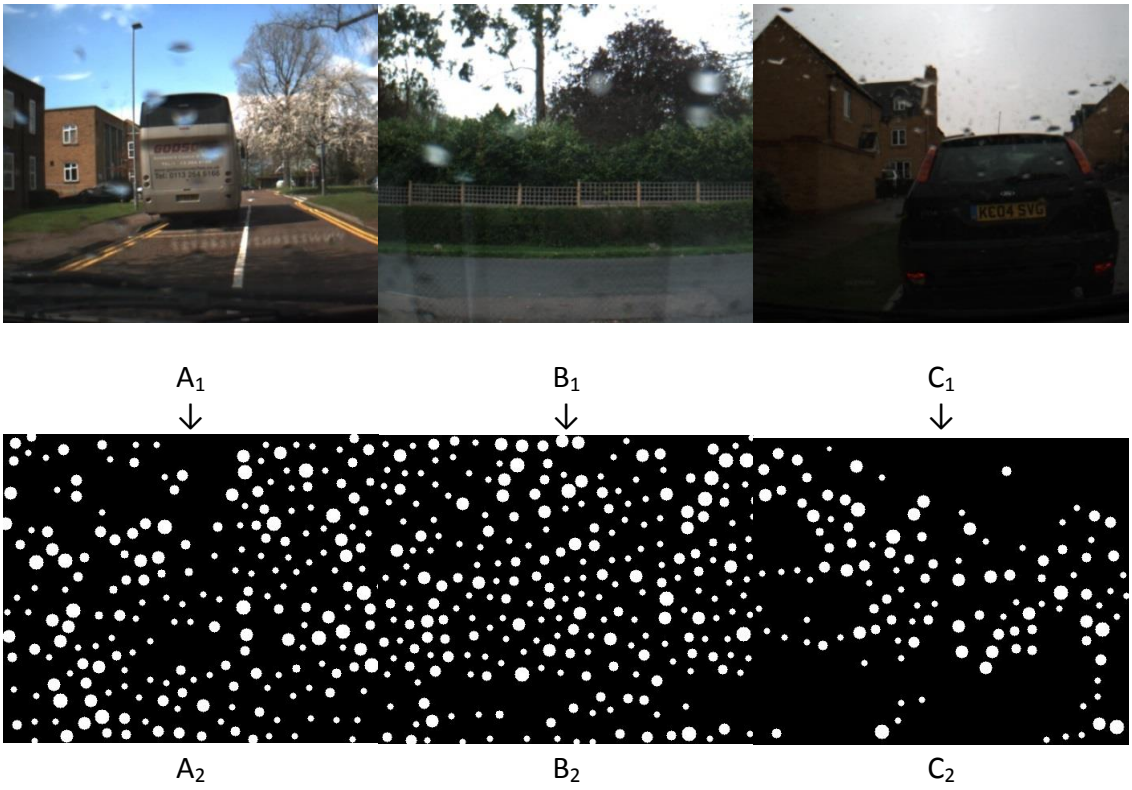


Figure 19 – Exemplar images after Hough Circle Transform

Figure 19 shows that, even in instances when raindrops have a clear defined edge (Figure 19 C_1), or that the intensity difference between the drop and the background (Figure 19 A_1, B_1), the Hough Circle Transform does not collate with the location of the raindrops. As this type of feature detection is computationally

expensive, these results raise the question of if this feature detector brings any benefit to the saliency model.

3.3.5 Weighted Sum of Feature Maps

All feature maps that are generated in sections 3.3.2 (*I)*, 3.3.3 (*T*)and 3.3.4 (*S),* are then summed using the ideal weights taken from [1] to produce an overall feature map F_{map},

$$F_{map} = 0.2232I + 0.315T + 0.4611S$$

Equation 4 – Addition of feature descriptors

This results in a combined saliency map, $F_{map,}$ as shown in Figure 20B. This image is then thresholded using an adaptive threshold [39] to give a binary mask, where regions of interest are non-zero, (Figure 20C) which can then be used to detect first stage candidate raindrops (Figure 20D). These identified candidate regions (regions that contain potential raindrops) are then formatted for use in the candidate classification process which is discussed further in section 4.5. Here the concept of the extended shape description of the raindrop is introduced (section 4.4) and the concept of isolating the raindrop from the context of the scene (section 4.3). Firstly, however we discuss the implementation the bag of word approach and why it is required.

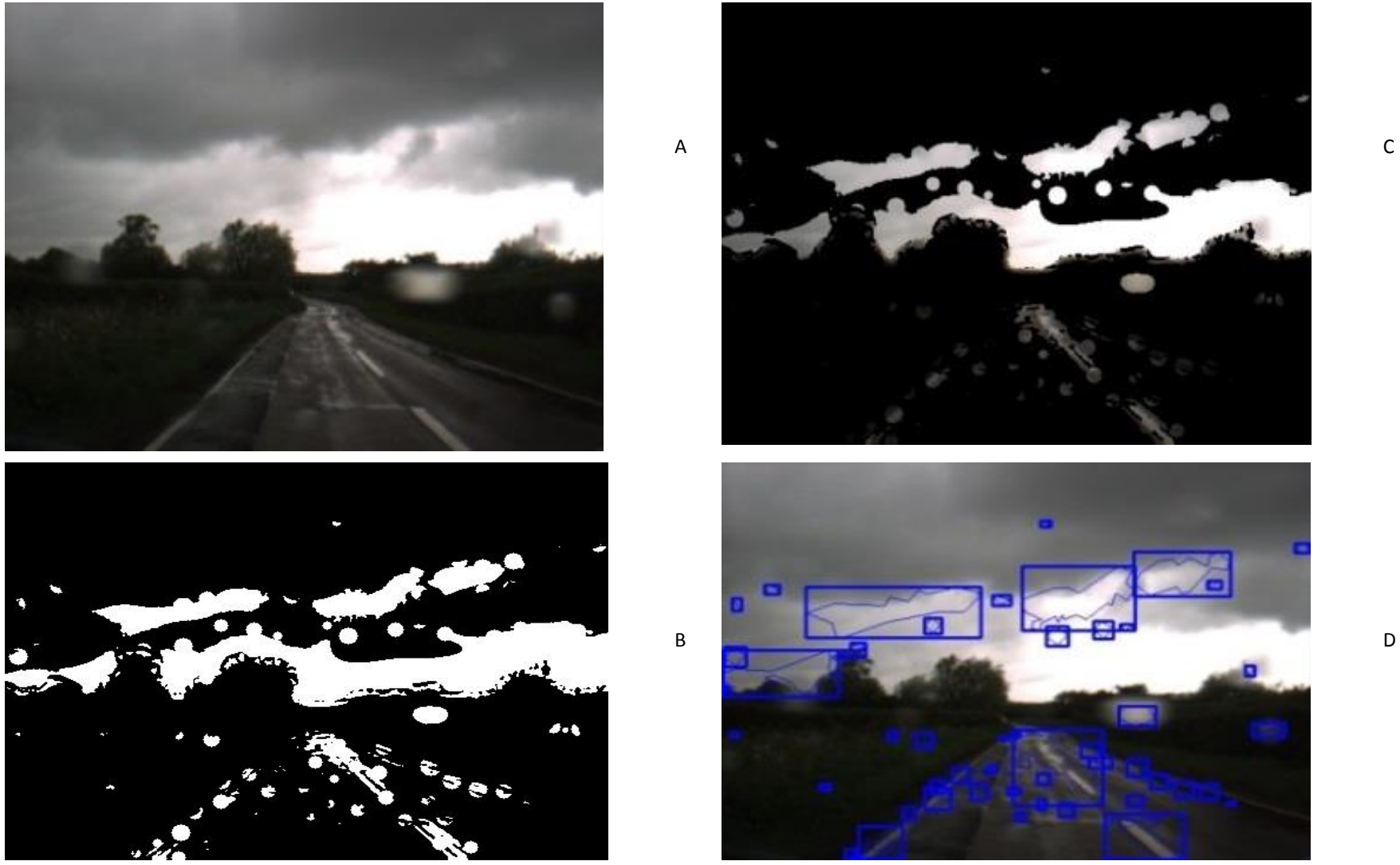


Figure 20 – Exemplar output of the all four saliency detection stages

4 RAINDROP CLASSIFICATION

Once the potential raindrops regions are identified by the saliency method described in Section 3, a means of verification has to be applied to confirm if the region contains a raindrop or not. As seen in Figure 20D, regions which do not contain raindrops can be erroneously identified by the saliency detector. To remove these regions would be computationally expensive, and could potentially introduce further errors within any system that may use the reconstructed image. Within this section, we introduce the concept of using a visual bag of words for feature representation (Section 4.1), clustering for reduction of the feature vector size for more efficient dictionary construction (Section 4.2) and how we generate our training set and how we classify our potential raindrop candidates using support vector machines and random forests (Section 4.5). We also introduce our novel concept of isolating the raindrop from the context of which it lies and extending the shape descriptor of the raindrop, and how these novel ideas extend the work of [1].

4.1 Bag of Visual Words

Usually used as a simplifying representation, the Bag of Words is used in natural language processing or information retrieval. Initially designed for text categorization purposed [40, 41], a bag of words can represent text, such as a sentence or document, where grammar and word order is disregarded but the multiplicity (or large number of conditions) is kept. This bag of words is a dictionary of all relevant keywords, or words that are relevant to the documentation class that is currently being used for training. There a many possible documentation classes, such as a children's book, or court case notes or spam emails. Based on this generated keyword dictionary, a vector representation can be generated whose size is the number of words in the dictionary and its components are the frequency of the appearance of the word within it. Therefore, each document that is to be queried can been described in terms of the appearance, or word count, of the keywords within the vector, or bag of words.

The use of this bag of words approach can be extended to computer vision, where a collection of visual words or "key points" [42] can be built. These key points represent the specific characteristics, or select features, of a particular region within an image. For example, if we were to select key points, or specific characteristics or a human face, we may pick the eyes, nose and mouth as these are key points of what defines a human face. These key points are translated as a numerical representation called a feature vector or feature descriptor, therefore giving us a bag of words containing numerical feature vectors, analogous to the bag of words. It should be noted that these key points, such as the raindrops, traffic cones and any other item described as

salient, must be invariant to any changes that can be applied, for example contrast, brightness, occlusions and geometrical transformations such as shift, scaling or rotation.

4.2 Feature vector quantisation / clustering

As it is computationally expensive and inefficient to build and query a dictionary constructed of all the training images, a quantisation algorithm is used to reduce the size of the vocabulary to a certain number of clusters. The use of K-Means [43] was selected for this. The rationale behind this against a variant such as Gaussian Mixture Models (GMM) [27] used by the authors in [1] was down to computational cost as the GMM algorithm's computational time exceeded what was deemed acceptable.

K-Means is a method originally developed for signal processing and is described as a method of unsupervised learning. Multiple feature points are clustered into *k* clusters, where each feature point belongs to the cluster with the nearest *mean*, thereby reducing high dimensional descriptors into a lower dimension by 'grouping' instances of similar attributes.

Cluster centroids: $c_1,\ldots,c_k$
Number of updates per cluster: $n_1,\ldots,n_k$
Initialise cluster centroids by randomly selected data points
$n_i = 1$ $(I = 1,\ldots,k)$
while *interation_count* < *max_iteration_count*
    select random data point: x
    find nearest centroid to x
        $t = argmin_{i = 1,\ldots ki,} \| x - c_i \|$
    update cluster centroid c_t
        $n_t = n_t + 1$
        $c_t = c_t + x + c_t / n_t$
end

Table 1 *k*-means clustering method adapted from [43]

For our application, an ascending cluster count of *k* was chosen from 20 (which is the original value in the work of [1]) to 80. These *k* points were randomly selected in the data space, and each pixel value was assigned to the closest cluster centroid. Once all these pixels were assigned, the positions of the *k* centroids were recalculated. This iteration was continued until the cluster centroids did not move (they stayed in the same x, y coordinates).

Based on the outline of the feature descriptors constructed, we end up with a set of 18 intensity features *Ii* (6 *(c,s)* scales pairing over all three XYZ colour channels), and similarly 18 texture features over the same (*Ti)*. This results in a 36 feature vector for each pixel for each identified raindrop candidate region (Figure 20D) detected in our

saliency detector method described in Section 3.3. Based on these features constructed from this described method we construct a visual bag of words representation using *k-means* clustering to construct a code word vocabulary upon which we will then construct visual code words for each given region via vector quantization. This follows the standard visual bag of word model for a fixed length histogram of feature occurrence [44] of dimensionality *k*. Figure 21 show visually how this vector was constructed.

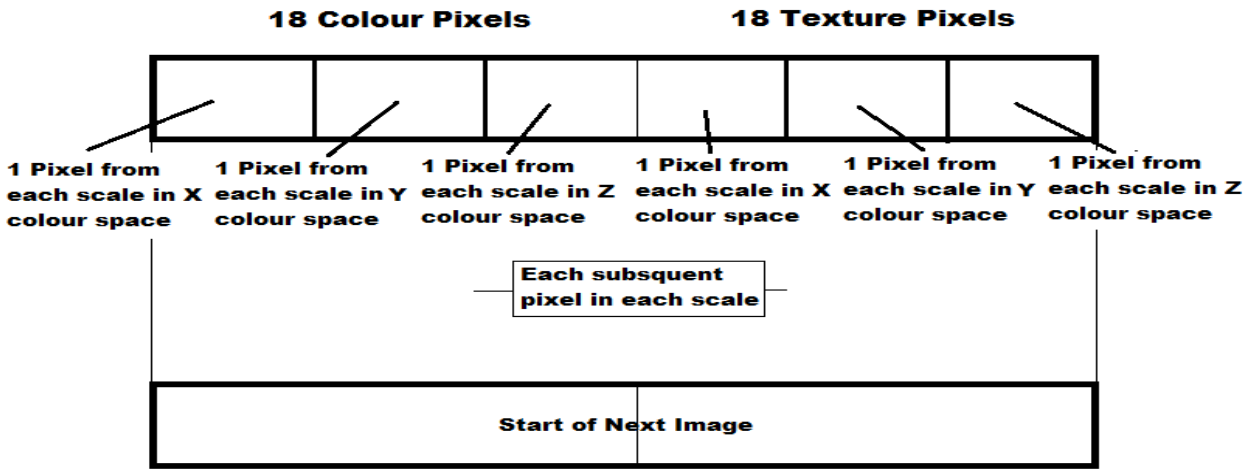


Figure 21 – Format of the pixel information normalisation.

Every pixel across every scaled pair over all three colour channels is added to this feature vector in the format shown in Figure 21. Each detected region containing a potential raindrop candidate as detected by the described saliency method (Section 3.3) is appended on to the bottom, thereby constructing a vector containing all pixel information for each detected region.

4.3 Raindrop Context Isolation

The idea behind implementing the novel approach of isolating the raindrop from the background image came from the notion that a raindrop should be able to be defined without any information of its surroundings. [1]

[1] Interference from the background on detecting objects has been seen, and common folk lore in Artificial Intelligence circles regales a story of a system being trained with images of tanks. One set of tanks was allied and the other set enemy [54]. However, as the context that the allied tank images were taken was vastly different from the enemy tanks images, one being day and the other at night, the machine, instead of identifying tanks, learnt the main contextual difference, that being night and day.

In our instance, isolating the raindrop from any background distractions means that we can be confident that the machine is seeing and learning the raindrop texture, colour and shape, and not the features of its surroundings. Thereby if a new background or context were encountered that the machine had not been trained against, it has a higher chance of correctly identifying the raindrop than if the context were still present. The added benefit of this approach is a reduction in the amount of pixels that require processing, and therefore computational cost, which is at a premium in an automotive environment, without losing the unique information required for classification.

We therefore extended the work of [1] which considers the entire bounding box of the candidate (e.g. Figure 20D), by considering a comparison of performance against only the region interior to the raindrop based on the thresholded raindrop area identified earlier. Firstly the contours of the raindrop candidate were isolate based on the work of [45]. These contours are then used as a mask. All pixels outside these contours were set to pixel value 0 (i.e. black) and the pixels within them remaining the same. The pixels that are not pixel value 0 are copied into a new image vector thereby isolating the raindrop from the varying scene context. Similarly, we utilise this approach on non-raindrop regions. This not only has the same benefits as already described, but means that the resulting shape is vastly different from the raindrop candidate region. This means, that with the context removed, the machine has a stronger chance of correct identification as raindrops are vastly different from non-raindrops. Using this approach within our secondary verification significantly differs from the prior work of [1].

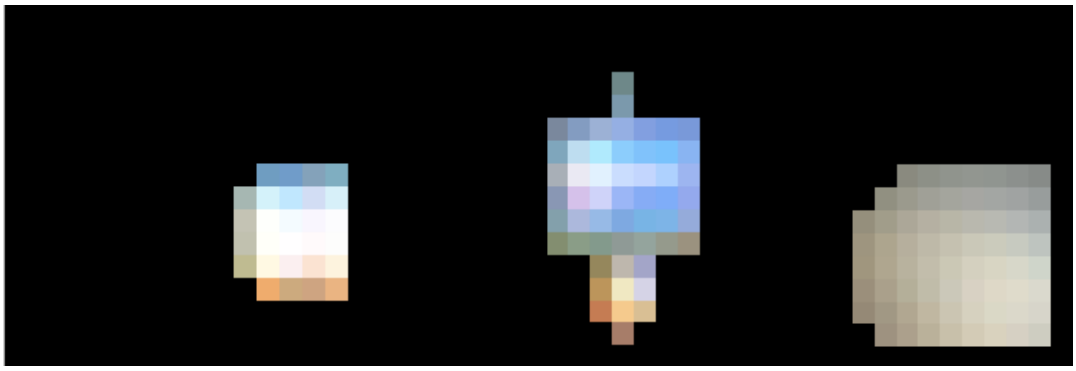


Figure 22 – Three exemplar images of isolated raindrops

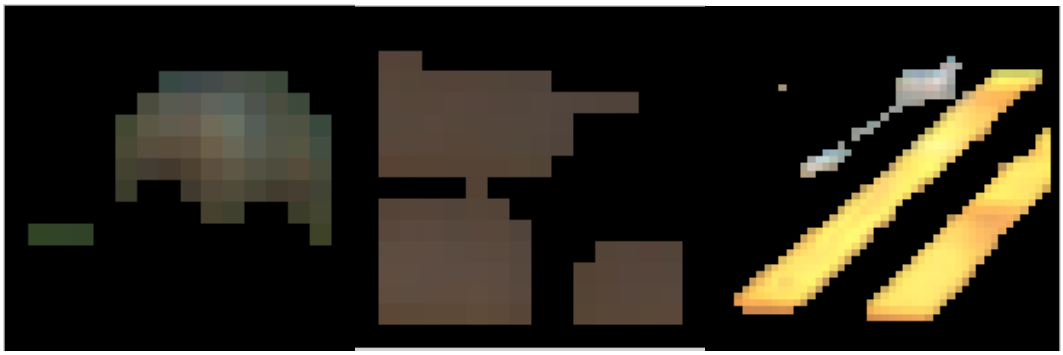


Figure 23 – Three exemplar images of isolated non-raindrop objects

Figure 22 and Figure 23 show examples of isolated raindrops and non-raindrop objects. It can be seen that as the shape, colour and texture of the raindrop is vastly different from the non-raindrop objects, isolating this information from the context creates a greater emphasis of these differences. It should be noted that the black pixels evident in the example images are not present when being added to the visual bag of words, as they are ignored during the copy.

4.4 Extending the Shape Descriptor Information

Due to the limitations already described in using the Hough Transform for detecting circular raindrops (Section 3.3.4), we investigate the notion of extracting a further extended shape descriptor of dimensionality of seven based on the seminal Hu Moments [11] using the equation where n is the normalised moment,

$$
\begin{aligned}
I1 &= n_{20} + n_{02} \\
I2 &= (n_{20} + n_{02})^2 + 4n_{11}^2 \\
I3 &= (n_{30} + 3n_{12})^2 + (3n_{21} - n_{03})^2 \\
I4 &= (n_{30} + n_{12})^2 + (n_{21} - n_{03})^2 \\
I5 &= (n_{30} + 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] \\
&\quad + (3n_{21} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2 \\
I6 &= (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03}) \\
I7 &= (3n_{21} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] - (n_{30} - 3n_{12})(n_{21} \\
&\quad + n_{03})[3(n_{30+}n_{12})^2 - (n_{21+}n_{03})^2]
\end{aligned}
$$

Equation 5 – Rotation invariant moments (Hu Moments)

As Hu Moments are rotation, translation and scale invariant, this adds a stronger shape descriptor compared to the Hough circle transform alone. As seen in Figure 18 B_1, raindrops can have a very similar shape, even though they may be smaller and / or rotated, and as described in Section 4.3 raindrops have vastly differing shape characteristics to non-raindrop objects. Therefore using the moments of the raindrop and non-raindrop gives us a more accurate shape description.

We extracted and calculated the 7 seminal Hu Moments [11] for each identified raindrop, by first extracting the contour of the raindrop using of structural analysis [45] of the thresholded candidate region (Figure 20C). This provided the contour on which we calculated a set of seven invariant shape measures (invariant to rotation, translation and scale) as an additional novel shape feature for our overall raindrop detection and verification approach.

4.5 Feature Classification

Based on the visual bag of words vocabulary that we create in Section 4 and our Hu moment based shape descriptor (Section 4.4) we have an overall combined feature descriptor, or feature vector, of length *k + 7* where *k* is the number of clusters used in our earlier *k-means* derived visual bag of words model (Section 4) and *7* is the number of Hu moments calculated (Section 4.4). These feature vectors are then used in both feature training and feature classification within a machine learning (ML) algorithm. Two machine learning algorithms were chosen for use within this thesis, giving us a means to investigate the differing performance of different (ML) algorithms for this particular classification problem of raindrop detection. Those chosen were Support vector machines (SVMs) [46] to follow the techniques used in [1] with our new novel techniques and Random Forests (RF) [10], which is significantly different in computational manner to SVM's.

4.5.1 Support Vector Machines

A support vector machine is a supervised learning, discriminative classifier defined by a separating hyperplane, or in other words the algorithm outputs an optimal hyperplane after training which can be used to categorise new candidates depending on which side of this hyperplane it falls on.

4.5.1.1 Linear Example

If we were to take a set of points, in Cartesian space for simplicity, which represent our candidates and had a clear device between them, we can separate these by a line.

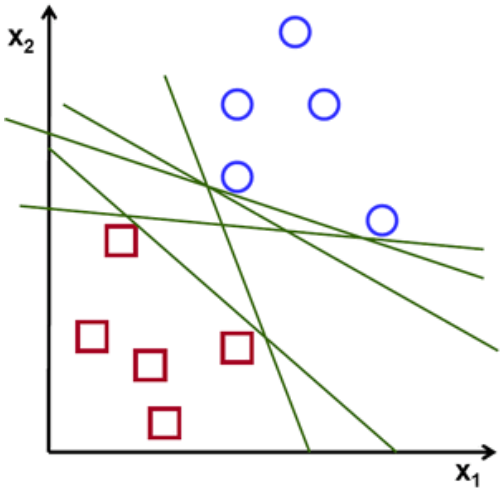


Figure 24 - Line separation in Cartesian space of candidate examples [47]

An example of this line can be shown in Figure 24, where a line can clearly divide the candidates into clear classifiers (i.e. two distinct groups either side of the line). However, as seen in Figure 24, the dividing line can intersect at varying places. Therefore, a method of which line is 'best', or give the clearest separation between differing candidate groups needs to be implemented. It can be side that a line is bad if it passes close to a point because it will be noise sensitive, it can be further insinuated that the best fit line should pass as far as possible from all points. This is called the Maximum Margin, where the SVM algorithm attempts to fine the largest minimum distance to the training examples, maximizing the margin of the training data.

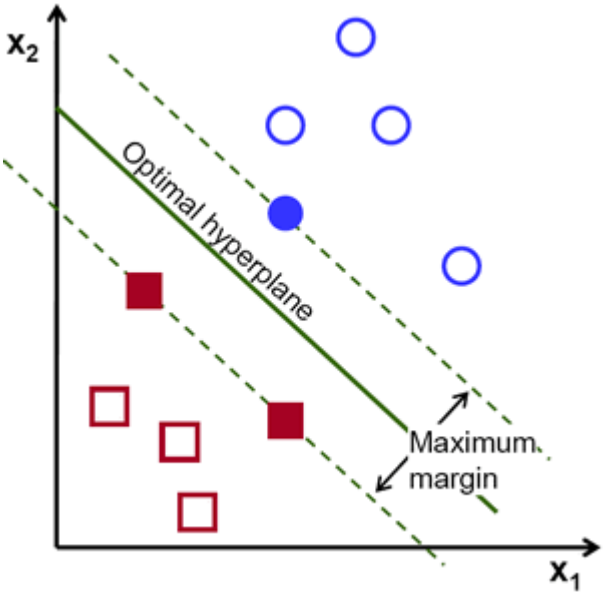


Figure 25 - The optimal hyperplane after finding the maximum margin [47]

Figure 25 shows the best fit line after finding the maximum margin of all the possible hyperplanes. It should be noted, the maximum margin is used as this has a small chance of misclassification, due to the small error in the location of the boundary. It is also robust against removal of any non-support-vector data points and empirically it works very well at most classification problems. When a new, unseen candidate is presented to the classifier, it attempts to ascertain which side of the hyperplane this example falls upon. Of course, not all points fall neatly between a linear separator, which is where non-linear SVM's excel.

4.5.1.2 Non Linear SVM example

In the instance where data is not linearly separable, or cannot be defined by way of a binary solution, we can project the data to a higher dimensional space where it becomes separable. Non-linear SVMs are employed and fit a 3D plane is between the projected data points. The best way to this of this is a pile of coloured beads of two colours lying on the floor. Projecting these beads from the 2D space (as in on the floor) to 3D space would be akin to making all one colour of beads stick to the roof and the other colour remains on the floor. Similar optimisation takes place to then find the best fit 3D hyperplane separator. Figure 26 shows this process where the plane is projected to a higher space, separating the two colours of beads. The light circle is the 2D line where the 3D hyperplane intercepts with the projected plane. These hyperplanes can be projecting into more than 3 dimensions, however this is challenging to visualise.

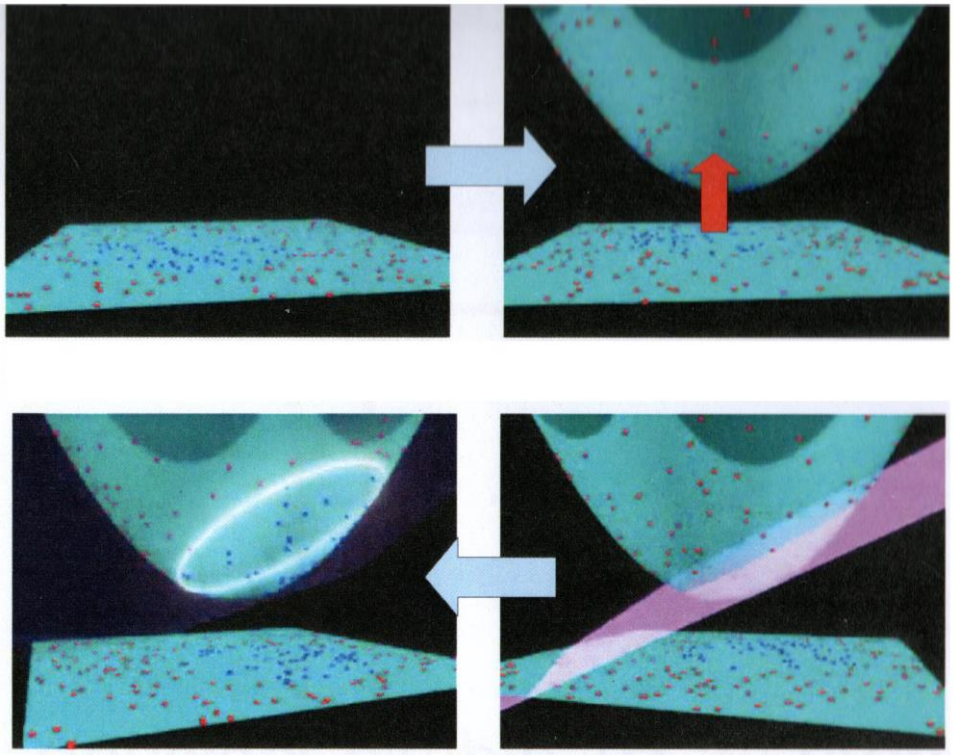
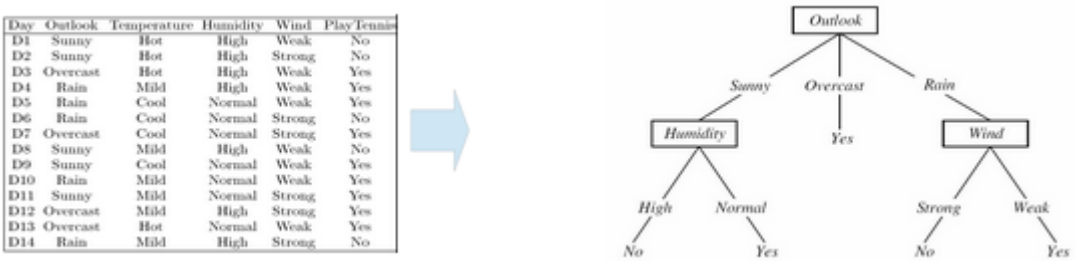


Figure 26 - Projecting 2D plane to 3D to separate candidate data. The white circle in the last image is the 2D line where the 3D hyperplane intersects with the projected plane. Images taken from video [48]

4.5.2 Random Forests

Random forests are described as a supervised learning algorithm, much like the SVMs (section 4.5.1); however this is where the similarities end. Random forests are constructed of multiple decision trees, where an input is fed in the top of our tree and travels down, getting bucketed into smaller and smaller sets until a decision is derived.



Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Figure 27 - OK to play tennis [49]

As the example in Figure 27 shows, the tree advises us if it is ok to play tennis depending on observed weather conditions. If we were to consider that the outlook was sunny, for example, and the humidity was observed to be normal, the decision would be that the conditions are met to allow us to play. A random forest takes this

one step further by constructing multiple decision trees in to an ensemble. Ensembles are a group of weak learners, our decision trees, combined into a strong learner which is our random forest[2].

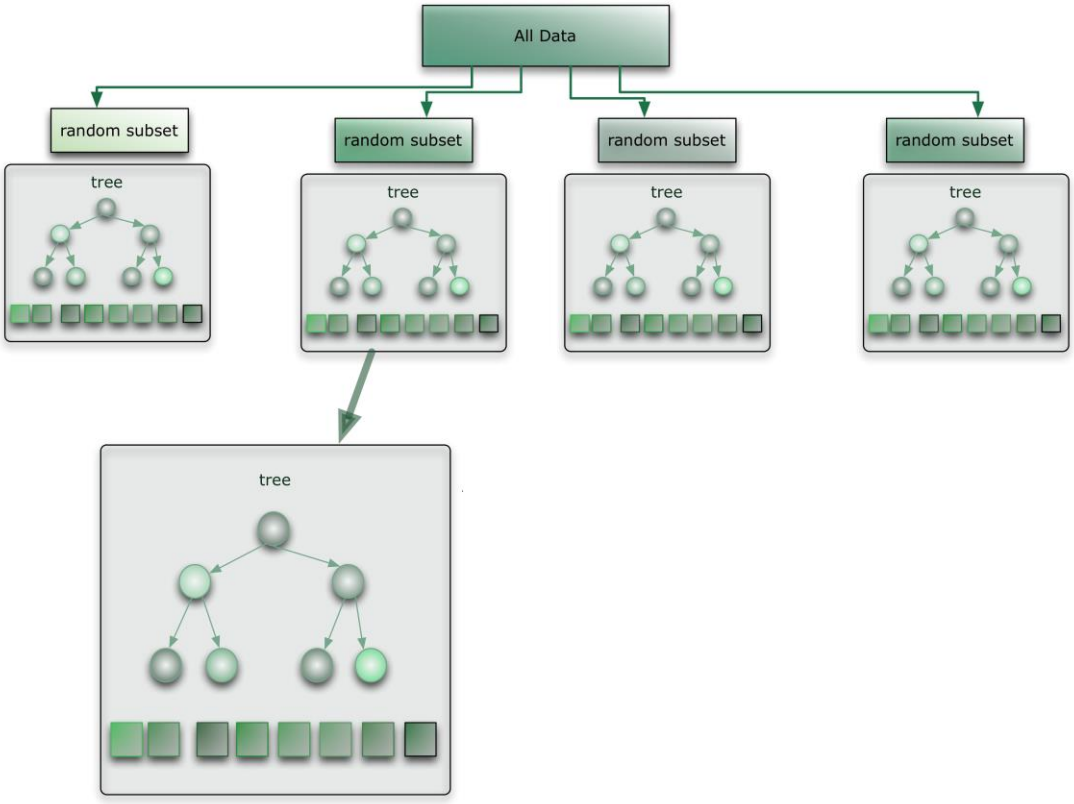


Figure 28 - Example ensemble of decision trees making a random forest. Images adapted from [50]

Figure 28 shows the construction of a random forest containing an ensemble of decision trees. Training the random forest involves selection a sample of *N* cases (outlook, wind, etc.) at random, therefore creating a subset of the data. This data is the training set for the growing tree. A number *m* is specified such that at each node, *m* variable are selected at random from *M* input variables (sunny, overcast, etc.). Each tree is then grown to the specified size. In our case, this was variable and performance measured. Other variables altered were the maximum depth a tree could grow (how many decision trees attached to other decision trees), and the variable *m*.

Once training is completed, the random forest is queried by entering in a new, unknown candidate at the top of the tree. The candidate then runs down all of the trees and an average or weighted average is taken of all of the final nodes that are

[2] It should be highlighted that ensembles in machine learning are not limited to decision trees and random forests, indeed, we could consider an ensemble of SVMs, where these machines (trained on different data sets) are shown the same image and a mean or weighed sum of all the outputs are taken to decide the classification

reached. Due to the nature of the random forest, runtimes are fairly fast, and in some cases, parallelisation can be utilised to speed up decision times further. However, in some cases, where the data is particularly noisy, these machines can 'over-fit' datasets. This occurs when the machine starts to 'fit' or train to the data given, instead of training to the underlying relationship.

4.5.3 Classifier Training

Our Support Vector Machines (SVM) and Random Forests (RF) were trained on feature descriptors for both positive and negative raindrop examples. SVMs were trained based on the Radial Bases Functions (RBF), linear kernel (KN) and the Polynomial kernel (PK), using a grid search over the kernel parameter space. RF [10] were trained over a set of varying parameters including a max depth *d* from 2 to 25 in increments of 2, sample count s_c from 5 to 30 in increments of 5, and maximal tree count t, from 100 to 2000 in increments of 100. Training was conducted against every permutation of all these parameters and the best performing presented for critical comparison here.

4.5.4 Generation of Training and Testing Image Dataset

The training image dataset was initially generated by manual selection of regions containing both raindrops and non-raindrops across various road types (i.e. Dual Carriage, Country, Urban) and various rain conditions (heavy, light, non-rainy). To further extend our images dataset for both raindrops and non-raindrops, we transposed by 90 and 180 degrees, adjusted brightness / contrast to both brighten and darken the image set, and mirror inverted the image. This gave us a larger dataset with potential training images that were not captured and therefore giving a more robust training set.

The image dataset used for testing the classifiers was generated in a similar fashion; however different regions were selected from different images throughout our recorded dataset. Images were again manipulated in the same fashion as the training dataset, giving us a testing dataset that differed significantly resulting in a fair test of the trained classifiers.

5 RESULTS

Within this section we analysis the results by providing tables with best and worst performance highlighted, for true positives, false positives and performance, recall and accuracy. We first look at how the data set was gathered, and the formulas used that generate the figures shown in the results. We then present the results both qualitatively and within real world images, first returned from the method used by the authors in the work of [1], the same method but with the raindrop context isolation only (as described in 4.3), using the same method but with Hu Moments only (described in 4.4), and the same method but with both raindrop context isolation and Hu Moments. We then provide a comparative overview of all methods, isolating strongest and weakest performers, with possible explanation on the cause of this outcome.

5.1 Experimental Setup and Data Gathering

All test video data was gathered using a forward facing Point Grey 'Bumblebee' stereo digital camera (1024x768) mounted behind a car windscreen (Figure 29). This was considered as the location of most vehicle camera vision systems are located, and therefore gives us an accurate simulation of how these systems might see raindrops. Footage was recorded under a variety of road environments, weather (rain) conditions (e.g. Figure 4), differing lighting conditions (e.g. light, dark, overcast, sunny) and different times of day. This resulted in a vast amount of representing our three previously defined raindrops in Section 1.2 across varying different types of conditions for use in training and testing.

Figure 29 - Camera location in car

5.2 Analysis of Results

Within the raindrop feature descriptor and classification approach outlined, we consider the results from the best performing identified parameter for a given cluster count (*k*) (from 20 to 80) for both SVM and RF. We indicate True Positives (*TP*), True Negatives (*TN*), False Positives (*FP*) and False Negatives (*FN*) against four variants – the prior colour/texture descriptor approach of [1], the same but with the use of the aforementioned raindrop context isolation and separately, with the use of Hu Moments shape descriptor, and finally all of these combinations. For further consideration, recall (*r*), precision (*p*) and accuracy (*a*) are also plotted by means of the following equations to give an overview of the machines strength in those particular parameters.

$$p = \frac{tp}{(tp + fp)}$$

Equation 6 – Precision metric

$$r = \frac{tp}{(tp + fn)}$$

Equation 7 – Recall metric

$$a = \frac{(tp + tn)}{(tp + tn + fp + fn)}$$

Equation 8 – Accuracy metric

5.3 Presentation of Results

Results are presented in tabular form, giving a quantitative indication on the performance of the differing methods described in the previous sections. We also present the results using frames taken from 'real world' images where the raindrops fall into one of the three categories previously defined in section 1.2: undefined raindrops, defined raindrops, and no raindrops. This provides a qualitative indication of the performance supplementing the quantitative analysis of both the RF and SVM machine algorithms across the various raindrops encountered. Results are also plotted as graphs in APPENDIX C – GRAPHICAL RESULTS showing the trends that were observed.

5.4 No Raindrop Context Isolation or Hu Moment Information

Results given in this section state the performance of the previous given methods first described in the work of [1], in which we do not implement any raindrop context isolation, nor any feature descriptor extensions based on the Hu Moments. This gives us a baseline of the performance of the initial investigation. Table 2 show a collated analysis of the results, with the best and worst performers highlighted.

5.4.1 Quantitative Results

Within Table 2 we can see a favourable TP outcome for *k*=20 (clusters) and the SVM classification (TP = 79%); however, we also suffer notable high false positive outcomes (FP = 87%). The TN is low at 13% and similarly FN is 14%. We can see a general trend where increasing *k* (clusters) has a detrimental effect on the performance of the SVM classification performance (falling TP) from an overall high of 79% at *k*= 20 to 46% at *k*=80. TN is seen to a favoured over TP when (*k*) = 80 (clusters), where previously the SVM favoured the TP. The FN rate however increased from 28% at *k* = 60 to 50% at *k*=80 and the FP rate falls by 23% showing a shift in the machine preference when increase the feature space.

By contrast the RF classification performance of TP is seen to increase by 4% to a maximal detection rate of 64% at *k*=60 (Table 2). However increasing *k* results in a decrease of TN from the maximal value of 78% at *k*=20 (clusters) with the lowest recorded at 70% at *k*= 40. An increase in FP is observed when increasing *k* from a minimal value of 22% at k=20 to the maximal recorded value at 30% at *k*=40 reducing to 28% at *k*=80. FN is shown to remain stable over the increasing cluster centres with a

maximal FN rate of 35% at *k*=20 to a minimal value of 31% observed at both *k*=40 and *k*=60 (clusters)

The precision, accuracy and recall for the RF and SVM are calculated using the equations stated in Section 5 (Equation 6, Equation 7, Equation 8) over increasing values of *k*. The highest recall is at 86% with *k*=20 whilst using SVM, however both precision and accuracy is at 48%, affected by the high FP. The highest accuracy was recorded at 65% at *k*=40; however there is negligible impact observed when increasing *k*. A maximal value of accuracy is observed at 72% when using RF at *k* =20, with negligible differences being observed through increasing *k*.

Although initially TP is high at 79% (Table 2) when using SVM at *k*=20, FP is unacceptable at 87%. Increasing the cluster by 20 does show to have a negative impact on recall, but accuracy and precision both have a marked increase as the cluster count (*k*) is increased, with negligible impact observed thereafter. The ideal cluster count is at *k*=40 for this problem, giving the best all round in recall (68%), precision (63%) and accuracy (65%). RF accuracy is affected by increasing the cluster count (*k*), however over all the ideal *k* value for this solution is observed to be 60, giving the best all round performance with regards to recall (69%), precision (69%) and accuracy (70%).

Classifier @ cluster count	True Positive	True Negative	False Positive	False Negative	Precision	Recall	Accuracy
RF k=20 d=24 sc=25 t=100	60.24%	**77.93%**	22.07%	**34.91%**	**73.19%**	65.09%	**71.76%**
RF k=40 d=18 sc=20 t=1300	63.45%	70.13%	**29.87%**	31.44%	67.99%	68.56%	**69.37%**
RF k=60 d=14 sc=5 t=1900	**63.56%**	71.08%	28.92%	31.32%	68.73%	68.68%	69.92%
RF k=80 d=22 sc=5 t=1700	62.57%	71.93%	28.07%	32.39%	69.03%	67.61%	69.85%
SVM k=20 kn=linear	**79.41%**	13.24%	**86.76%**	14.19%	**47.79%**	**85.81%**	**48.12%**
SVM k=40 kn=RBF	63.35%	62.22%	37.78%	31.55%	62.64%	68.45%	**65.21%**
SVM k=60 kn=RBF	66.56%	52.51%	47.49%	28.08%	58.36%	71.92%	61.84%
SVM k=80 kn=linear	45.87%	**76.09%**	23.91%	**50.44%**	65.74%	**49.56%**	63.34%

Table 2 - Tabulation of results for No Raindrop context isolation or Hu Moment Information

5.4.2 Qualitative Results

Figure 30 shows the output from both classifiers using the parameters outlined in [1] over increasing cluster counts (*k*). Blue borders denote where a raindrop was

identified by the classifier, and it can be observed that although the SVM appears to be fairly strong in identifying raindrop candidates when at k = 20 clusters, it does incorrectly identify some isolated regions as non-raindrops (middle left) when the region does contain raindrops and vice versa (top right). With regards to the distinct raindrops, although the salient operation successfully detects raindrop candidates, the majority are misclassified as non-raindrops, reflecting what was observed with the experimental figures in Table 2. The SVM classifier shows a stronger performance with the scene that contains no raindrops, accurately identifying the majority of regions as non-raindrops, however a few misclassifications remain (top left). With increasing the cluster count (k) to 40 the SVM classifier fails to correctly classify any regions. This could be due to lack of suitable images given to the SVM during training, or that this codebook size is unsuitable for this problem. With an increase to k = 60, we can visibly observe the FN rate decrease compared to k= 20 and k=40 across all three test images. The regions previously misclassified (top right and bottom left) in the indistinct image are now correctly identified as non-raindrops; however some misclassifications remain (middle left). This increase in FN rate is observed in the distinct and no raindrops test images, where the majority of candidates are correctly identified; this is reflected in the test results shown in Table 2.

The RF classifier performs very poorly with these parameters, where all candidates were identified as non-raindrops across all values of k, with only one raindrop being identified in the distinct image when k=40 and k=60 (top left). This could be because of the lack of distinction between the images, and if so finding the correct tree may be impossible.

For this given problem, the SVM classifier proves to be the most accurate in correctly identifying raindrops and non-raindrops. Misclassifications do still remain across all three example image scenes; however increasing k from 20 to 60 does show an increase in accurately identified non raindrops.

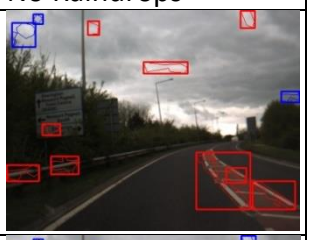
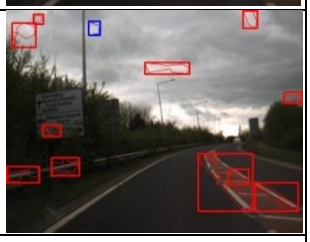
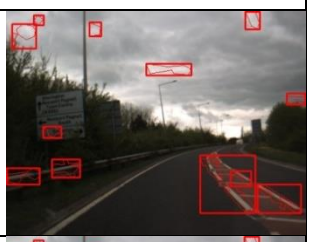
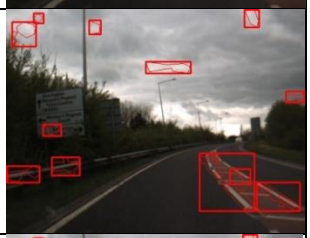
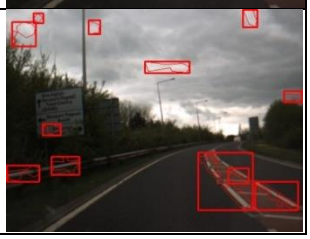
ML	*k*	Indistinct	Distinct	No Raindrops
SVM	20			
	40			
	60			
RF	20			
	40			
	60			

Figure 30 – Output from classifiers using parameters outlined in [1]

5.5 Raindrop Context Isolation / No Hu Moment Information

Results given in this section state the performance of the previous given methods first described in the work of [1] along with isolating the raindrop from the context of the scene as described in Section 4.3. The feature descriptors are not extended in this

instance. Table 3 shows a collated analysis of the results, with the best and worst performers highlighted.

5.5.1 Quantitative Results

Table 3 shows the impact of isolating the raindrop pixels from the environmental context and following the approach of [1] (without Hu Moments shape descriptor). We observe the TP and TN detection rate increases using the SVM classifier over the results presented in Section 5.4.1 with TN showing the greatest increase from the minimal value at 13% (Table 2) to 73% (Table 3) where *k*=20. TP shows an increase with the maximal value of 73% at *k*=60, up from 67%. A decrease in FP and FN (falling FP and falling FN) rates are observed with FP falling by 60% at *k*=20, but with an increasing FN of 11% compared to Table 2 at the same cluster count (*k*). Further decreases in FP and FN can be observed with around 10% reductions being seen for FP detection rates and marginal decreases for FN across all cluster counts *(k*) in comparison to the results presented in Table 2

The RF shows a marginal increase in detection rate for TP in Table 3 to 79% at *k*=80 compared to Table 2 (63%), however a marginal decrease in TN is observed on the majority of cluster counts (*k*). A decrementing effect can also be observed on the FP detection rate showing an increase of 6% at *k* = 20 (clusters) and 3% at *k*=80 compared to Table 2. However the FN detection rate decreases by 18% at *k*=80 (14% compared to 32%).

The recall and accuracy are observed to be higher when using RF compared to using the SVM, with marginal differences being observed with increasing *k*. As can be seen from Table 3, when using RF the highest recall (86%), precision (73%) and accuracy (78%) can be seen at *k*= 40 with marginal differences over increasing *k*. SVM shows the highest recall (79%) at *k*=60, precision (77%) at *k*=80 and accuracy (76%) at both *k*=40 and *k*= 60.

In this configuration, the RF favours identifying raindrops showing, a higher TP than the SVM with the same parameters, however it delivers the worst FP overall. With regards to the TN, the opposite is true, where this is favoured by the SVM, however delivering the worse FP compared to the RF. Table 3 indicates that *k*=40 gives the best overall performance in recall (85%), precision (73%) and accuracy (78%) when using RF for this given problem. SVM is not as strong as performer as the RF, showing that RF is better suited for this problem.

Classifier @ cluster count	True Positive	True Negative	False Positive	False Negative	Precision	Recall	Accuracy
RF k=20 d=10 sc=5 t=1600	78.43%	**71.61%**	28.39%	**15.26%**	73.42%	84.74%	77.92%
RF k=40 d=14 sc=15 t=1200	**79.45%**	71.12%	28.88%	14.15%	73.34%	**85.85%**	**78.20%**
RF k=60 d=10 sc=20 t=1400	78.88%	71.33%	28.67%	14.77%	73.34%	85.23%	78.01%
RF k=80 d=12 sc=5 t=1800	79.24%	68.82%	**31.18%**	14.38%	**71.76%**	85.62%	**76.89%**
SVM k=20 kn=RBF	69.84%	72.60%	27.40%	24.53%	71.82%	75.47%	73.98%
SVM k=40 kn=linear	66.07%	79.45%	20.55%	28.62%	76.27%	71.39%	75.57%
SVM k=60 kn=linear	**73.41%**	72.18%	**27.82%**	20.68%	72.51%	79.32%	**75.61%**
SVM k=80 kn=linear	56.57%	**82.63%**	17.37%	**38.88%**	**76.50%**	**61.12%**	**72.29%**

Table 3 - Tabulation of results with Raindrop context isolation and no Hu Moment Information

5.5.2 Qualitative Results

Figure 31 shows the output of the classifier on detected regions using only raindrop context isolation. It can be observed that the SVM classifier failed to correctly identify any regions across all three example image scenes for all values of *k*. Indeed, all first stage detected regions are incorrectly classified as raindrops, except for in the indistinct scene when *k*=40. Here, two regions containing raindrops are misclassified (red border). Removing the context in this instance has a vast detrimental effect on the SVM classification accuracy.

The RF classifiers performance is similar to the performance of the SVM. Indeed, the RF classifier misclassifies all raindrops in all three example image scenes when *k*=20 or *k*=40, however performance notably increases at *k*=60, with few misclassifications evident. This is in contrast to the experimental results that show the classifier performance plateauing at *k*=40. However, it is observed that for this configuration, the RF classifier does outperform the SVM, backing up the experimental results for precision, accuracy and recall in Table 3.

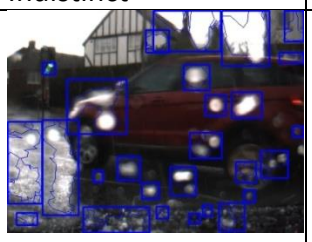
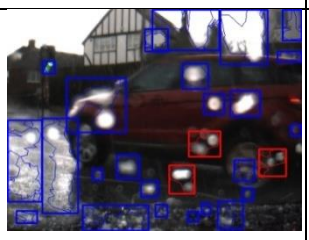
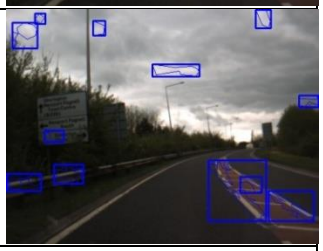
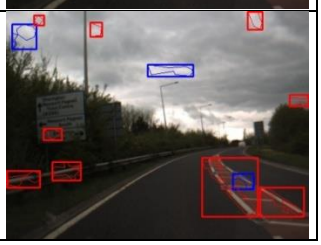
ML	*k*	Indistinct	Distinct	No Raindrops
SVM	20			
	40			
	60			
RF	20			
	40			
	60			

Figure 31 - Example of classifier output on detected regions using raindrop context isolation

5.6 Hu Moment Information / No Raindrop Context Isolation

Results given in this section state the performance of the previous given methods first described in the work of [1] along with extended the raindrop feature descriptor using Hu moments using the method described in Section 4.4. The raindrop is not isolated

from its context in this instance. Table 4 shows the collated analysis of the results, with the best and worst performers highlighted.

5.6.1 Quantitative Results

Results in Table 4 show an increase in both TP and TN detection rates for both SVM and RF classifiers compared to the results shown for previous parameters. The SVM in this case has a TP of 69% for *k* =80, up from 46% in Table 2 (No Hu moments or context isolation) and 57% in Table 3 (context isolation only) for the same *k* values. However an overall decrease in TP is observed compared to the maximum detection rate from the initial configuration (79%, Table 2 – SVM *k*=20). TN shows an increase to 93%, the strongest detection rate compare to both previous configurations (76% and 63% for No Hu moments and No context isolation, and context isolation only, respectively) for the same *k* values. A decrease in FP to 7% and a decrease in FN to 26% for *k* = 80 is observed compared to Table 2 and Table 3 for the same *k* value.

RF shows a moderate increase in TN to 77% at *k* = 80, up from 72% for No Hu moments or context isolation (Table 2) and 71% for context isolation only (Table 3). A FP of 23% at *k*=60 is observed, a decrease from 29% for both No Hu moments or context isolation and context isolation only at the same *k* value. A FN of 25% is observed at *k*=20, this is a decrease from 35% observed in the parameters for Table 2, but an increase of 10% compared to context isolation of the raindrop in Table 3 for the same value of *k*.

Across both classifiers, the precision, recall and accuracy exhibit minor differences over increasing *k* with the performance of both classifiers plateauing at *k*=40 and *k*=60, highlighting poor performance in this lower dimensional feature space (smaller visual words codebook). However, overall we can see the top performing results with the addition of Hu Moments out performing those of the prior state of the art of [1] and having a marginal impact over context isolation only.

Classifier @ cluster count	True Positive	True Negative	False Positive	False Negative	Precision	Recall	Accuracy
RF k=20 d=14 sc=25 t=1800	**69.63%**	71.96%	**28.04%**	24.76%	**71.29%**	**75.24%**	**73.54%**
RF k=40 d=24 sc=5 t=1900	68.89%	74.89%	25.11%	25.56%	73.29%	74.44%	74.67%
RF k=60 d=22 sc=20 t=1700	68.54%	**77.47%**	22.53%	25.94%	75.26%	74.06%	**75.83%**
RF k=80 d=22 sc=25 t=1800	67.55%	77.30%	22.70%	**27.01%**	74.84%	**72.99%**	75.22%
SVM k=20 kn=POLY	**69.03%**	91.60%	**8.40%**	25.41%	89.15%	74.59%	**83.42%**
SVM k=40 kn=POLY	68.43%	92.87%	7.13%	26.06%	90.56%	73.94%	83.77%
SVM k=60 kn=POLY	67.90%	92.87%	7.13%	**26.63%**	90.49%	73.37%	83.50%
SVM k=80 kn=POLY	68.82%	**93.04%**	6.96%	25.64%	**90.82%**	74.36%	**84.06%**

Table 4 - Tabulation of results with Hu moment shape description and no raindrop context isolation

5.6.2 Qualitative Results

Figure 32 shows the resulting classifier output on detected regions using only additional Hu moment shape descriptors, and it can be observed that the SVM classifier favoured this configuration. Although misclassifications are apparent in the indistinct example image scene, with the contrast of the sky causing difficulties, the majority of classifications are correct. This can be observed across the two other examples image scenes, where misclassifications in the no raindrop image scene can be attributed to the nature of the shape of the identified region, where the Hu moments closely resemble a raindrop. The classifier remains stable across all values of *k*, shown by all detected regions in all three image scenes having the same classification output.

The RF classifier performs poorly in this configuration, incorrectly identifying all detected candidate regions are raindrops until *k* =60, where a shift occurs to the majority of regions being misclassified as non-raindrops with a few regions being correctly identified. This can be observed in the indistinct image scene where three regions are correctly identified as raindrops by their blue border. Although not as apparent, this shift from misclassification of raindrops to non-raindrops can be seen in Table 4 where FP falls from 28% at *k*=20 to 23% at *k*=80 and FN rises from 25% at *k*=20 to 27% at *k*=80.

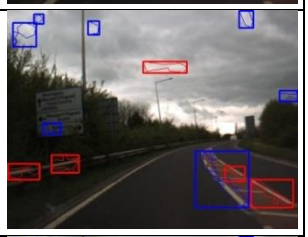
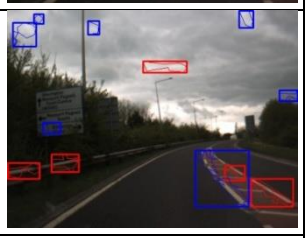
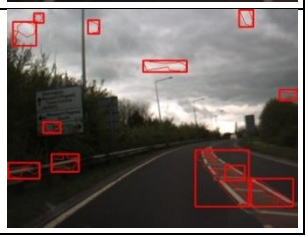
ML	*k*	Indistinct	Distinct	No Raindrops
SVM	20			
	40			
	60			
RF	20			
	40			
	60			

Figure 32 - Example of classifier output on detected regions using Hu moment shape descriptors

5.7 Raindrop Context Isolation and Hu Moment Information

Results given in this section state the performance of the previous given methods first described in the work of [1] along with extended the raindrop feature descriptor using Hu moments using the method described in Section 4.4. The raindrop is also isolated from the context of the scene as described in Section 4.3, thus providing us with an

overview of the performance of the system using both methods. Table 5 shows a collated analysis of the results, with the best and worst performers highlighted

5.7.1 Quantitative Results

Table 5 shows the greatest increase in TP and TN detection rates for both classifiers against the prior state of the art [1] (Table 2), and with using raindrop context isolation and the Hu moment shape descriptor individually (Table 3 and Table 4 respectively).

The SVM in this case has a TP of 79% at *k*=80, which is an increase of 6% from the previously observed highest detection rate of 73% (Table 3 – SVM, *k*=60) and a TN of 93% at *k*=40 (clusters), which was also observed when only using Hu moment shape descriptors and no context isolation (Table 4 – SVM, *k*=80). FP and FN show a decrease to 14% and 7% respectively, with the FN figure being the best observed, down from 21% to 14% (Table 4 – SVM, *k*=60).

RF classifier shows a TP of 80% at *k*=60 and *k*=80 and a TN of 77% at *k*=60. These results are similar to the results from both using raindrop context isolation only (Table 3- RF, *k*=40, TP), and Hu moment shape descriptors only (Table 4 – RF, *k*=60, TN) showing that context isolation works well for identifying raindrops, and using Hu moment shape descriptors is more suited to identifying non-raindrops when using the RF classifier. This is also reflected in the FP and FN, which shows a 2% decrease from the previous high recorded value of 23% using only Hu moment shape information (Table 4 –RF, *k*=60) and a decrease of 1% decrease when only isolating the context (Table 3 – RF, *k*=80).

Classifier @ cluster count	True Positive	True Negative	False Positive	False Negative	Precision	Recall	Accuracy
RF k=20 d=8 sc=10 t=1900	77.05%	**79.38%**	20.62%	**16.75%**	78.89%	83.25%	81.24%
RF k=40 d=20 sc=15 t=1900	78.64%	77.44%	22.56%	15.03%	77.70%	84.97%	81.06%
RF k=60 d=24=sc=5 t=1200	79.70%	77.19%	**22.81%**	13.89%	77.75%	86.11%	**81.48%**
RF k=80 d=16 sc=15 t=1200	**80.26%**	72.99%	27.01%	13.28%	**74.82%**	**86.72%**	**79.59%**
SVM k=20 kn=linear	68.47%	92.30%	**7.70%**	**26.02%**	89.89%	**73.98%**	**83.50%**
SVM k=40 kn=POLY	76.87%	92.83%	7.17%	16.94%	91.47%	83.06%	88.14%
SVM k=60 kn=POLY	78.85%	92.90%	7.10%	14.80%	91.74%	85.20%	89.20%
SVM k=80 kn=POLY	**79.17%**	**93.15%**	6.85%	14.46%	**92.04%**	85.54%	**89.49%**

Table 5 - Tabulation of results with both Raindrop context isolation and Hu Moment information

5.7.2 Qualitative Results

Figure 33 shows the classification results for given regions when using both context isolation and Hu moment shape descriptors. The SVM exhibits a similar behaviour to Figure 32, where the classifier stabilises at *k*=40 and shows no observable difference when increasing the cluster count further. Few misclassifications remain with the sky causing a misidentification of a raindrop, and the shape and colour of some the selected raindrop candidate regions (middle left) causing issues resulting in a misclassification of a non-raindrop. Further training using these and similar features could potentially result in correct identification of these detected regions. There are also apparent misclassifications in the no raindrop test image, with the same classification output given across all values of *k*. These appear to be caused by similar confusion in the classifier as in the indistinct test image, where colour and shape descriptor information closely resembles a raindrop. As the results are identical across all values of *k*, further training with more example images of similar shape and size is suggested. The stabilisation of the machine can be observed in Table 5 with minute changes to accuracy, recall and precision with differing *k* clusters.

RF exhibits poor performance when *k*= 20, however it does improve when cluster counts are increased, showing similar results to the SVM classifier. Some misclassifications still exist across all example image scenes, with one raindrop in the indistinct example continuously misclassified. This misclassification could be due to the shape of the raindrop exhibiting a shape descriptor more alike to a non-raindrop than

to a raindrop. The contrasting regions in the sky also cause some confusion with the classifier when analysing the non-raindrop scene, however these are detected as raindrops, again due to the shape descriptor and the colour of these detected regions being more similar to raindrops than of non-raindrop candidates.

The results in the images above can be related to the experimental values reported in Table 5, where FP rises and FN decreases over increasing *k* for RF and FN decreases for SVM but FP remains relatively stable. This is most evident in the indistinct example scene when using the RF classifier, where changes to the decisions made by the classifier can clearly be seen.

ML	*k*	Indistinct	Distinct	No Raindrops
SVM	20			
	40			
	60			
RF	20			
	40			
	60			

Figure 33 - Example of classifier output on detected regions using raindrop context isolation and Hu moment shape descriptors

5.8 Comparative Overview

Both classifiers shows significant improvements compared to prior art [1] (Table 2) with the additional use of our novel shape descriptor information (Hu Moments Section 4.4) and raindrop context isolation (Section 4.3).

Overall it can be seen (Graph 13, Graph 14) that the additional use of both Hu moment shape descriptors and raindrop context isolation results in the greatest

recall and precision (Graph 15, SVM). Few misclassifications still remain however; more evident in high contrast regions (e.g. gaps in the clouds and between buildings) when these regions are usually classified as raindrops, and where raindrops have combined, creating a Hu moment shape descriptor similar to a non-raindrop. These cases can prove difficult to overcome, due to the nature of how they appear to the classifier.

Experimental results do show that Hu moments aid in the reduction FP detecting rates and isolating the candidate from the context aids in the reduction of FN in both RF and SVM classifiers. As computational time is at a premium in automotive computer vision system, the reduction of FP and FN can be beneficial (less work needs to be carried out due to falsely identified regions). The opposite is true when looking at the TP and TN results. Isolating the candidate region was observed to produce a higher TP then when only using Hu moment shape descriptors. TN detection rates were aided by the addition of Hu moments, compared to only supplying the isolated pixels of the candidate. However both classifiers showed increases in detection rates for TP and TN and decreases in FP and FN when using both techniques, albeit marginally in some cases.

Increasing the cluster count when using all techniques does increase TP/TN and decrease FP/FN when using the SVM, and providing a larger code book in this instance helps classification. However the opposite is seen when using RF, where FP increases and TN decreases when k=40. Although FN does decrease in this circumstance, the reduction is marginal comparative to the increase of FP and decrease in TN.

With an automotive application in mind, reduction of FP and FN are crucial to reduce computational time, but the impact of missing a raindrop (reduction of TP) would need to be investigated further as this depends on the function that the system is tailored for.

5.9 Discussion on trends

During investigation of experimental results, a trend emerged on the detection rates when using the different classifiers with the various techniques. A greater increase in TP when isolating the raindrop from the context of the scene was observed when using the RF classifiers, compared to adding on Hu moment information. However, in the combined experimental results, only a 1% increase can be observed when using both context isolation and Hu moment shape descriptors when utilising the RF classifier (Table 5). Therefore adding the extra information in the form of Hu moments does not greatly improve the detection rates of the RF classifier in any meaningful way. Inversely, the SVM shows an

increase in TP when adding on Hu Moments (Table 4), although the increase is not as large as the observations of the RF (Table 3). This observation is believed to be caused by the unique way the classifiers work. The SVM uses the entire given feature vector information when training and during classification, the RF however, by its nature, uses random segments of the feature vector. Therefore, the chance of Hu moment information being selected for the random forest generation and classification are reduced. The SVM however, using the entire feature vector, prefers having the Hu moments, as these are very distinct for each given example shape, therefore creating a stronger classifier than pixel information on its own. Indeed, it may be possible to see similar results to the TN in Table 4 when only Hu moment shape descriptors are presented to the SVM.

5.10 Discussion on processing times

During testing it was found that to process a frame took in the region of 2 to 5 seconds depending on the number of raindrop candidates detected during the saliency phase. Tests were performed on a Core2Duo 2.1 GHz PC with 2 GB of memory with no specific optimisations for real time processing. These delays can be mostly attributed to the drawing routines that are currently used for our debugging and removing and optimising the code could give a significant boost in performance of the algorithm. It may also be possible to implement this code on a Field Programmable Gate Array (FPGA), giving a further increase in performance that may not be possible on a PC. FPGAs are programmable for a vast range of specific programs and can have dramatic increases in speed for image processing [51], indeed companies are beginning to investigate the different architectures being used, and are coming up with new technologies targeting the image processing market [52]. The benefit of using such a device would be twofold: First, it would be faster, as they would be application specific and therefore have fast frame rates, and secondly, have lower power consumption, therefore more suitable for in-vehicle use.

6 CONCLUSION

Within this work we present a novel combined feature descriptor for second stage raindrop verification, extending the proceeding methods detailed in [1]. We introduced two novel concepts in terms of the additional use of Hu moments based features and the use of raindrop context isolation where pixel information of the raindrop was extracted from any background scene pixel information. We also investigated the performance of two differing classifiers, Random Forests and Support Vector Machines, for classification and the effects of increasing cluster size, therefore producing a larger visual code book on these machine detection rates.

The work conducted produced a clear performance enhancement over the prior work [1] (Table 2 and Table 5) with only marginal impact over computation efficiency. The use of Hu moments in a combined shape and texture feature descriptor with the addition of raindrop isolation achieves a maximal 80% True Positive (TP) detection and 93% True Negative (TN) with marginal false positives and false negatives (Precision = 0.92). The impact of this work can be visually observed when using 'real world' example images from a typical rainy scene (Figure 33) This contrasts sharply with the prior state of the art [2,9,15] which suffered from high false positive detection (as illustrated in our comparative evaluation - Figure 30).

Consideration needs to be given to memory and computations costs however when increasing cluster counts or mathematical complexity of the discussed techniques. Certainly, within an automotive application, memory and processing time is a limited resource. Therefore results provided here have to be viewed with a critical eye with regards to cost of misclassification of a raindrop (potential missed / unusable frames later in the vision algorithm) or a misclassification of a non-raindrop (potential impact on computation and memory costs in the vision system).

6.1 Future Work

The work presented here does give improved results in terms of higher true positive and a significant reduction in false positives for raindrop detection, and from this base further work can be conducted. The use of primed visual word approaches, shown to be effective for alternative two-class problems [43], could be considered as a further enhancement. Further investigation into the use of increasing cluster counts could be conducted to investigate potential impact on detection rates for both SVM and RF. The utilisation of Hu moments could also be considered further for use with an SVM, reducing the memory and computational requirements for this given problem. Work considering the tolerance of detection systems would be also beneficial, finding a point when raindrop occlusion causes the system to fail to detect. Thus limiting the usage of such a system to when it is required, and therefore freeing system resources. As work has been conducted to gather stereo

images within this work, further work in this field may include using the resulting disparity maps to reconstruct the occluded region with the clear region from the sister image, and comparing the resultant images in a sign detection routine against typical infilling seen as in [23].

7 APPENDIX A – INVESTIGATION OF SALIENCY TECHNIQUES

During exploring the implementation into the technique chosen in the described paper[1], we investigated various established saliency techniques. The object was to see how well these techniques could highlight regions that contained raindrops and thereby simplifying the detection process defined in [1] and therefore reducing image processing time and memory use.

7.1 Blur, Sharpen and Subtract

The original image was blurred by using a Gaussian blur [29] and sharpened to create two images with the aim that after pixel by pixel subtraction, areas that contained little to no differences contained raindrops (as raindrops are bright blurred regions [8]) and therefore would contain no pixel value (i.e. region would appear black in resulting image). These steps are shown below in Figure 34.

Figure 34 –From left to right: Original image, blurred image, sharpened image and difference. As seen, not much difference is shown due to the fact that there is not much difference between the raindrops and the background.

This technique however produced poor results. As raindrops appear sharp in some instances, and other regions are blurred, the detection rate of likely raindrop candidates are outnumbered by the detection of many non-raindrop regions, thus resulting in a high processing time using second stage verification, and in some cases, the difference was not ample enough resulting in an image with no pixel values at all.

7.2 Sobel Image

The Sobel operator [26] was investigated due to its fast computational speed and it's slight resistance to noise due to Gaussian smoothing and differentiation being conducted using computation. The steps and outcome from this test is shown in Figure 35.

Figure 35 – From left to right: Original image and Image after Sobel operation. As seen, the raindrops are vastly visible on the second image.

As can be observed this gives a relatively strong indication of where raindrops are present, losing few a when the contrast difference is not great.

Figure 36 – From left to right: Original image and image after Sobel operation. As seen, although a few raindrops are visible, some blend into the background.

However, Figure 36 shows that when the contrast between the raindrops and background is not vastly different the results are not as strong. Although some raindrops are visible, there is no clear defined edge to them, thus making it hard to isolate them from the background. Although this image is challenging, further work could be conducted to see if this is useful when combined with other techniques.

7.3 Thresholding

As raindrops are described as bright regions on an image [8] thresholding [15] was tested on sample scenes to find if raindrops could be isolated from lower contrast regions of the image.

Figure 37 - From left to right: Original image and image after Thresholding

Figure 38 - From left to right: Original image and image after Thresholding

As seen in Figure 38, raindrops are not always white and in some instances have poor contrast relative to the background scene (i.e. when the scene is relatively dark). Lowering the threshold will result in too many false positives and having a detrimental effect on both second stage verification and speed of processing, already at a premium in automotive applications. Raindrops as shown in Figure 37 are detected well, however due to varying light levels seen in automotive computer vision, thresholding would be difficult to implement.

8 APPENDIX B – BAYES CLASSIFIER TESTING

As part of experimentation, Bayes classifiers [53] were tested alongside RF and SVM classifiers and detection rates were also recorded. The same test techniques (4.3, 4.4) were carried out on first stage detected regions (section 3.3) and results recorded for all True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Accuracy, Precision and Recall are also presented, along with a summary table of all results.

8.1 Quantitative Results

Table 6 shows that True Negatives (TN) is highest (90% when using Hu moments only) when using a lower cluster counts (*k*) which rapidly decreases when *k* (clusters) is increased (72% lowest value when using original technique in [1]). This is observed across all techniques under test. Indeed, we see marginal impact on TN detection rates when using the different techniques on test.

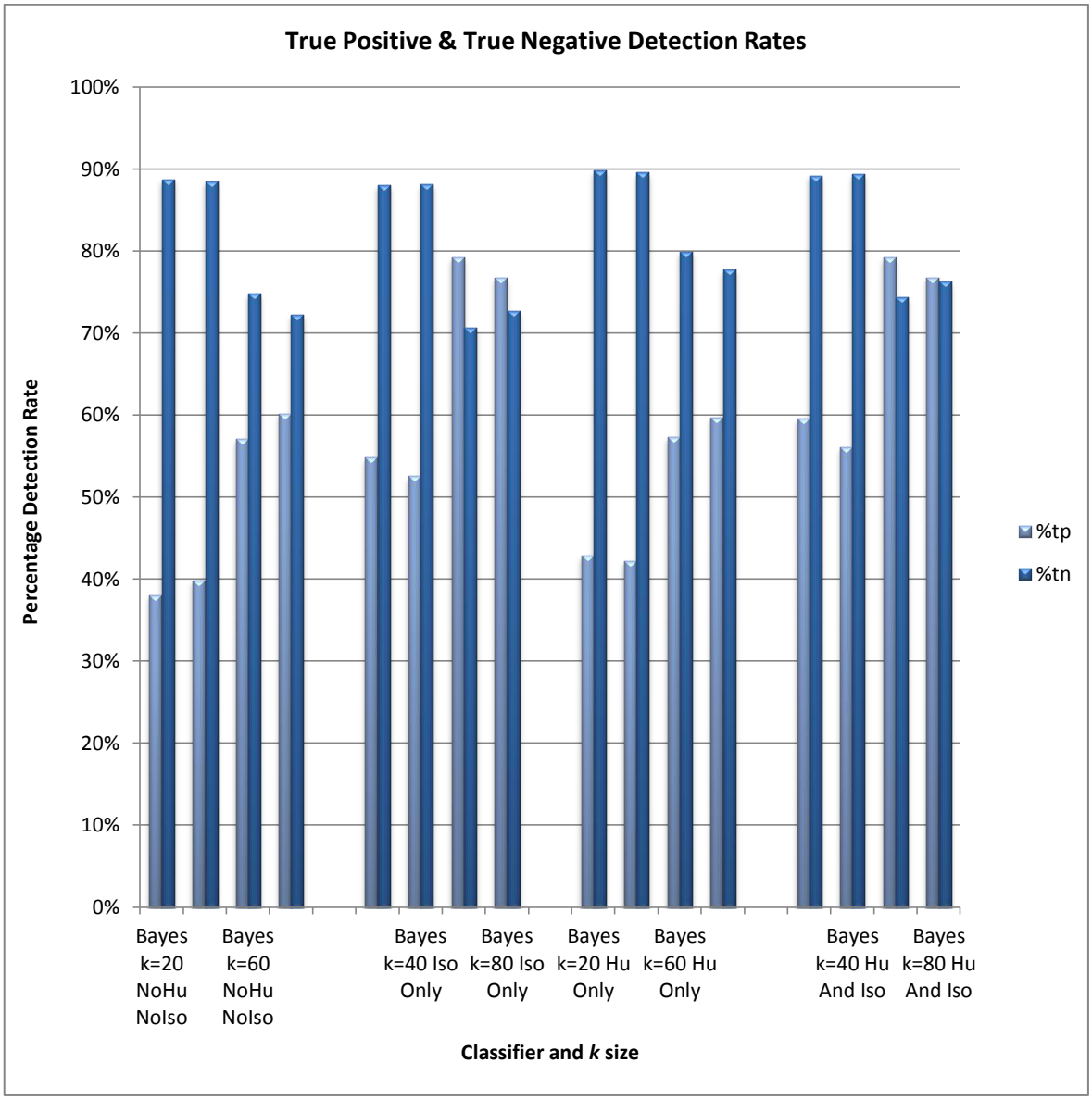
TP is observed to increase when increasing *k* (cluster) (38% to 60% when increasing cluster size for the original technique in [1]).Indeed at lower cluster counts the classifier is observed to be only slightly better than chance.

The FN is seen to decrease, in some cases up to 30%, when increasing the cluster count. This may be beneficial as the reduction of FN could reduce computational time when implemented in an automotive setting. It can also be observed that the different techniques have different impacts on FN values with using Hu moment shape descriptors and isolating the raindrop from the context giving an overall strong classifier when considering FP and FN detection rates. This can also be seen with regards to TP and TN where using all techniques and having a higher *k* value gives an overall stronger classifier. Precision, recall, and accuracy are highest when using both techniques and with a higher cluster count (*k*=80).

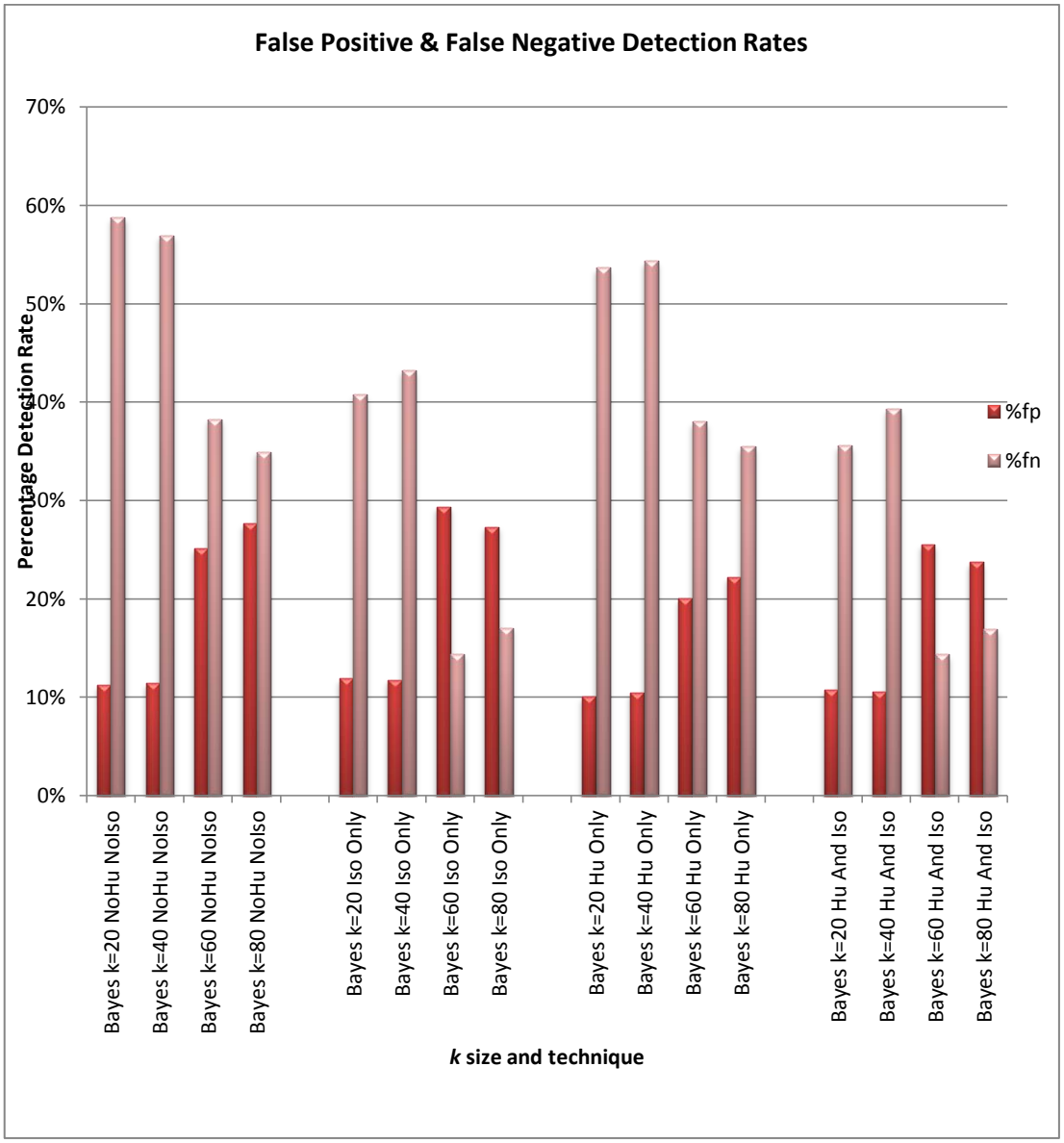
Therefore further work could be considered into increasing the cluster count which could produce a stronger Bayes classifier for this given problem.

Classifier @ cluster count	True Positive	True Negative	False Positive	False Negative	Precision	Recall	Accuracy
Bayes k=20 NoHu NoIso	38.10%	88.70%	11.30%	58.83%	77.13%	41.17%	65.85%
Bayes k=40 NoHu NoIso	39.83%	88.56%	11.44%	56.96%	77.69%	43.04%	66.68%
Bayes k=60 NoHu NoIso	57.10%	74.86%	25.14%	38.31%	69.43%	61.69%	68.53%
Bayes k=80 NoHu NoIso	60.17%	72.28%	27.72%	34.99%	68.46%	65.01%	68.79%
Bayes k=20 Iso Only	54.84%	88.10%	11.90%	40.75%	82.17%	59.25%	74.23%
Bayes k=40 Iso Only	52.54%	88.21%	11.79%	43.23%	81.67%	56.77%	73.10%
Bayes k=60 Iso Only	79.24%	70.62%	29.38%	14.38%	72.95%	85.62%	77.83%
Bayes k=80 Iso Only	76.73%	72.63%	27.37%	17.09%	73.71%	82.91%	77.57%
Bayes k=20 Hu Only	42.87%	89.90%	10.10%	53.68%	80.93%	46.32%	68.95%
Bayes k=40 Hu Only	42.23%	89.55%	10.45%	54.37%	80.16%	45.63%	68.44%
Bayes k=60 Hu Only	57.34%	79.94%	20.06%	38.04%	74.09%	61.96%	71.30%
Bayes k=80 Hu Only	59.71%	77.75%	22.25%	35.48%	72.86%	64.52%	71.39%
Bayes k=20 Hu And Iso	59.60%	89.23%	10.77%	35.60%	84.70%	64.40%	77.30%
Bayes k=40 Hu And Iso	56.11%	89.41%	10.59%	39.37%	84.12%	60.63%	75.57%
Bayes k=60 Hu And Iso	79.27%	74.44%	25.57%	14.35%	75.61%	85.65%	79.83%
Bayes k=80 Hu And Iso	76.84%	76.27%	23.73%	16.98%	76.40%	83.02%	79.52%

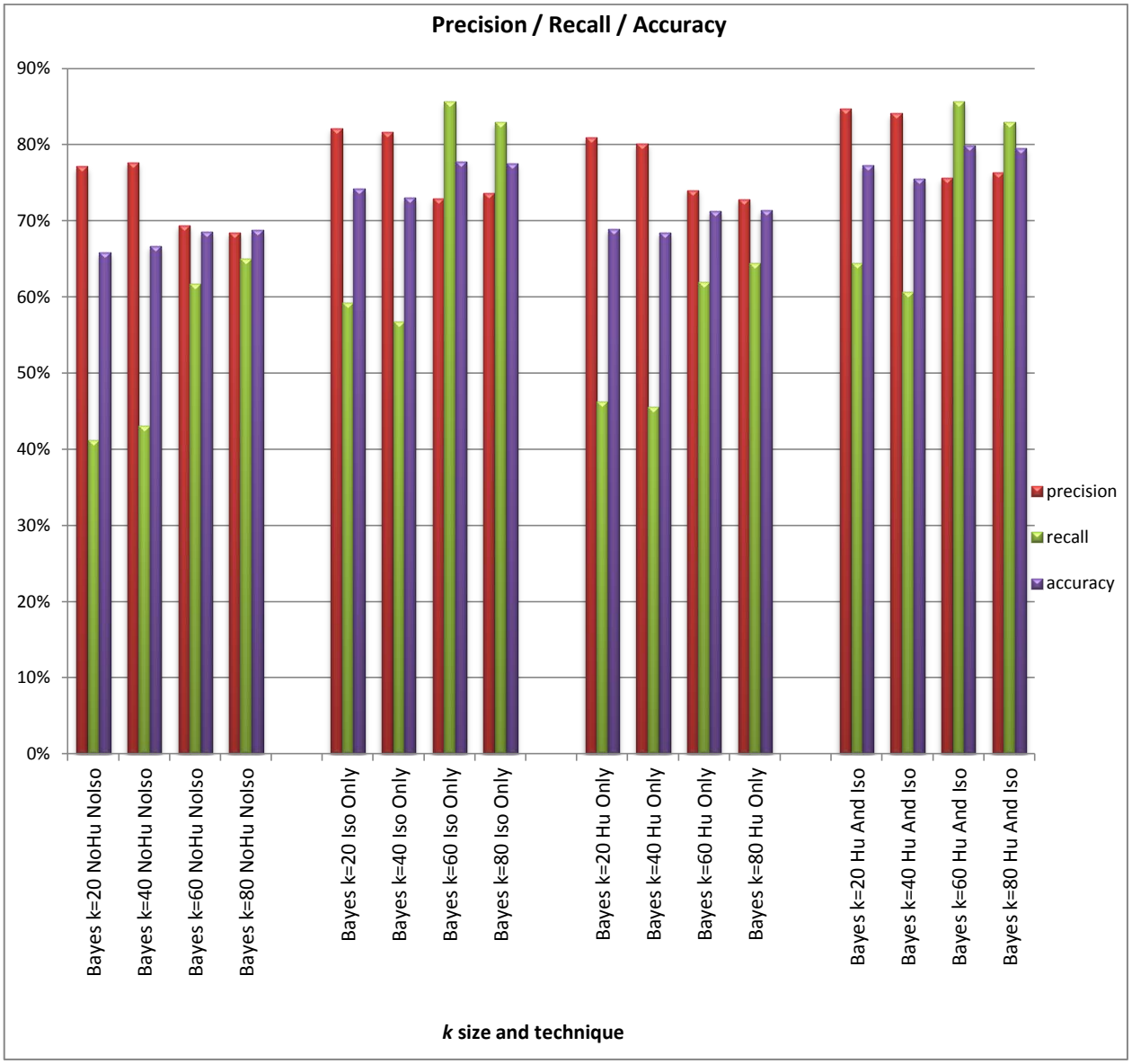
Table 6 - Tabulation of results with Bayes classifier for all techniques



Graph 1 – TP and TN experimental results with Bayes classifier for all techniques



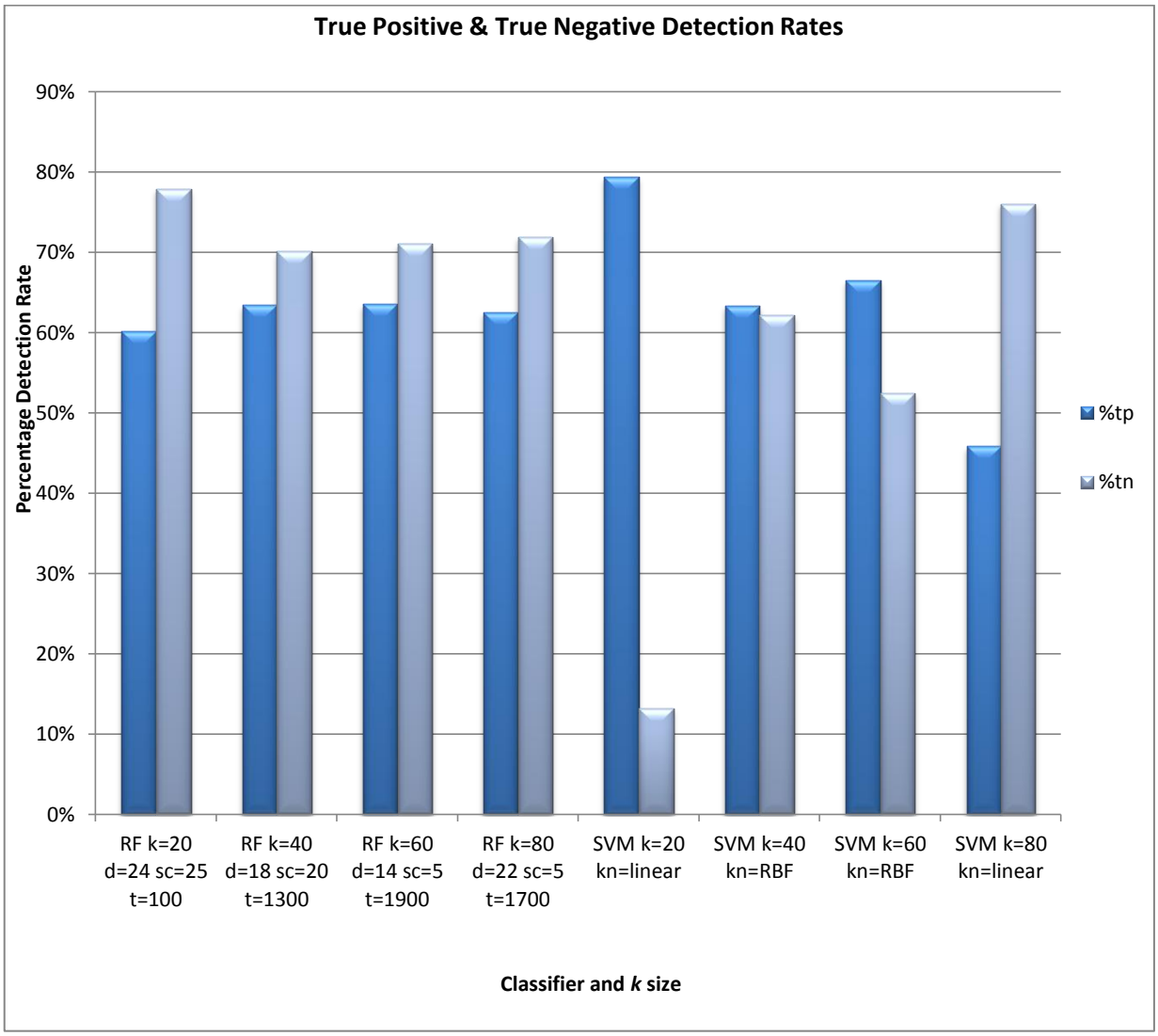
Graph 2 - FP and FN experimental results with Bayes classifier for all techniques



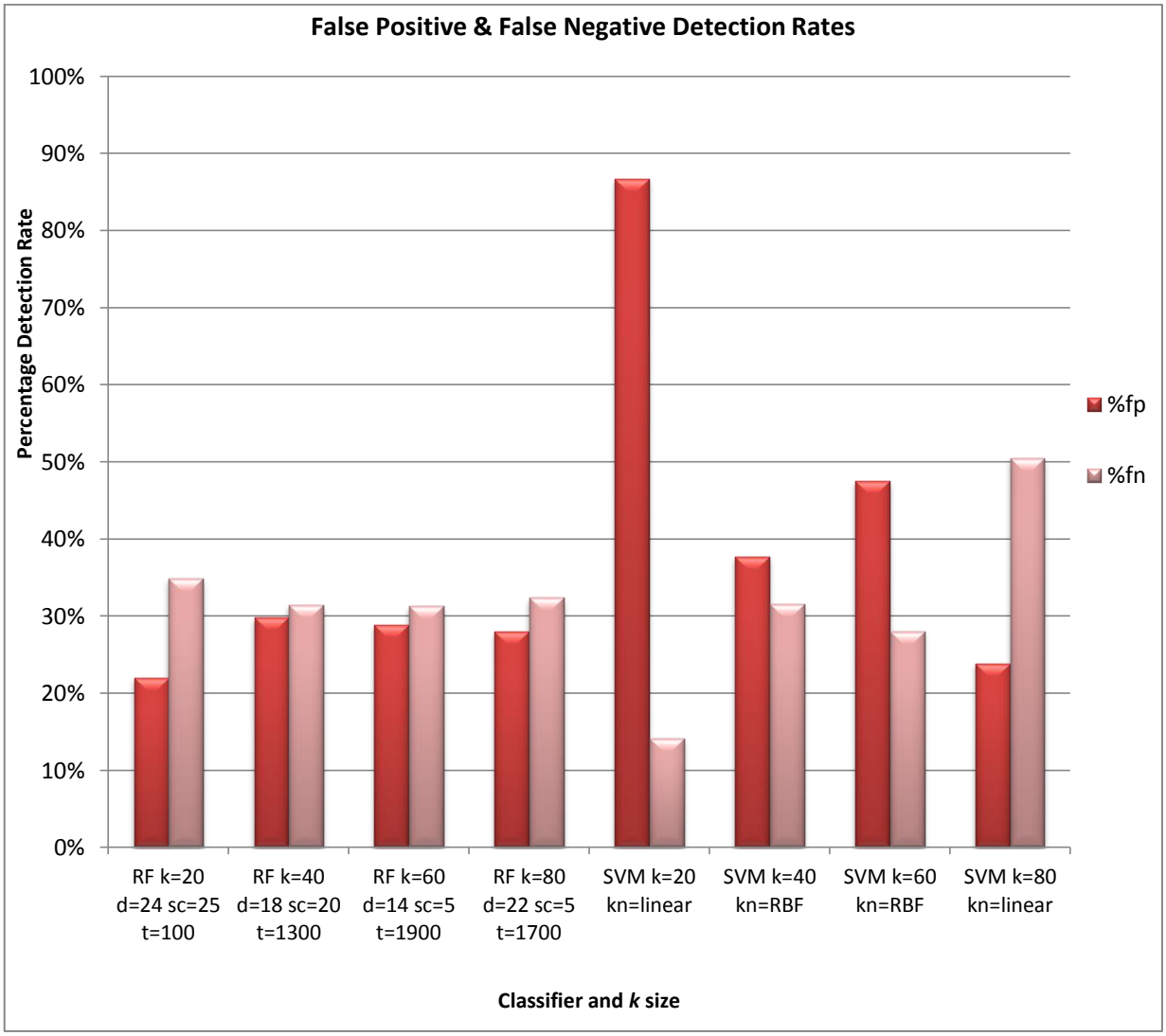
Graph 3 – Precision, recall and accuracy experimental results with Bayes classifier for all techniques

9 APPENDIX C – GRAPHICAL RESULTS

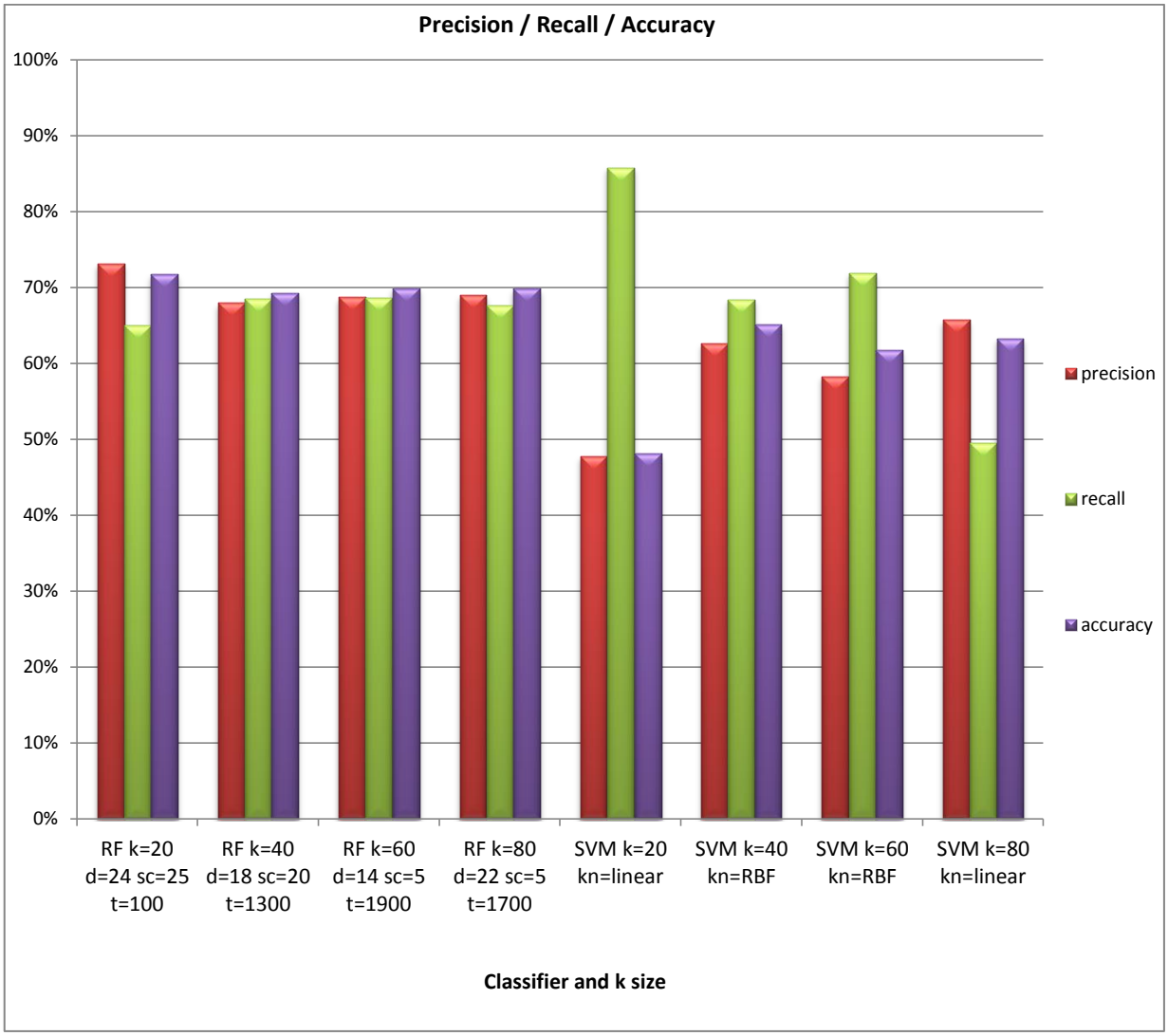
These graphs show the trends seen by using different combinations the configurations as described in Sections 4.2, 4.3, and 4.4 supplementing the tables shown in Section 5.



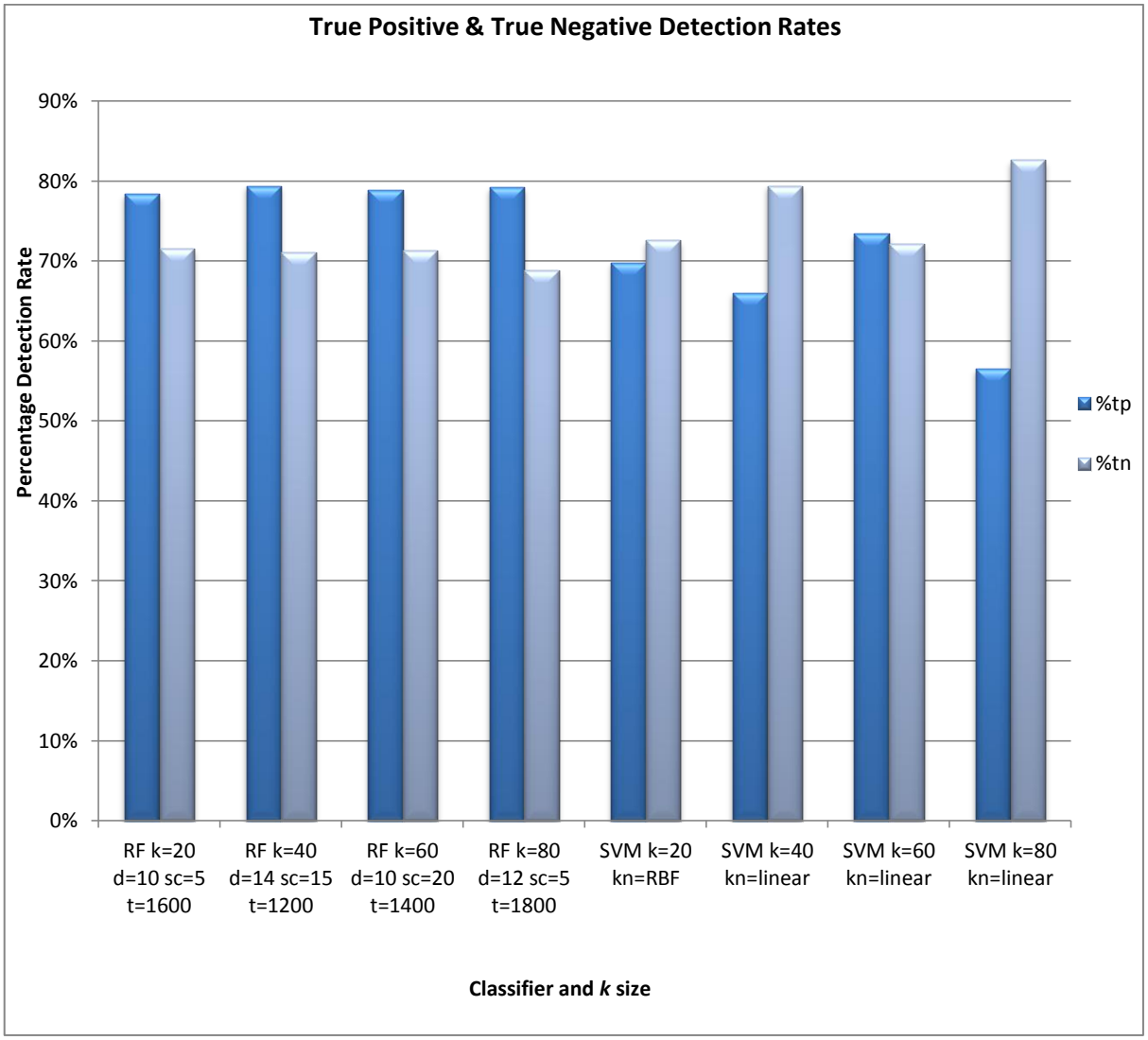
Graph 4 – True positive and true negative experimental results as outlined in [1]



Graph 5 – False positive and false negative experimental results as outlined in [1]

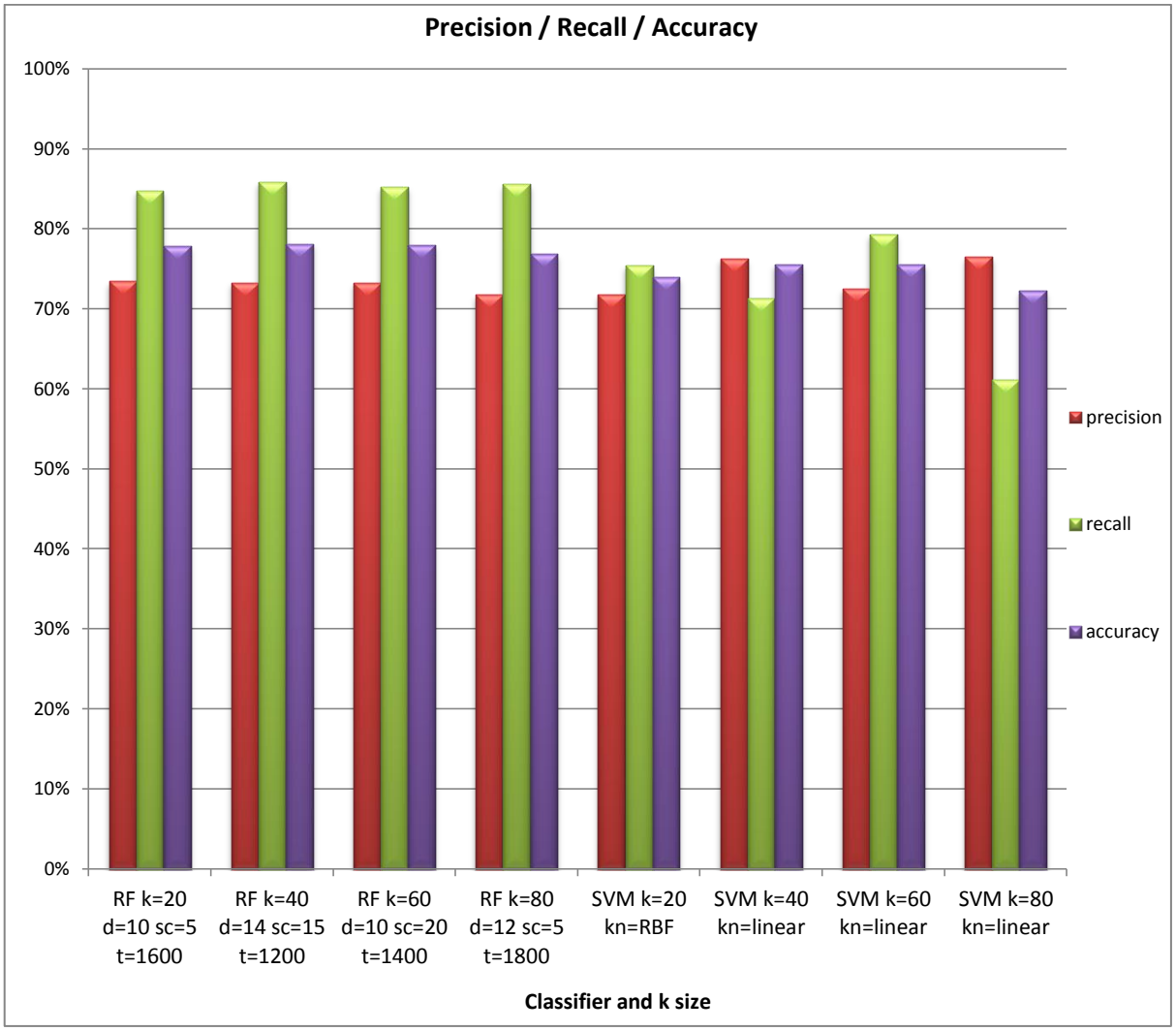


Graph 6 – Precision / Recall / Accuracy experimental results as outlined in [1]

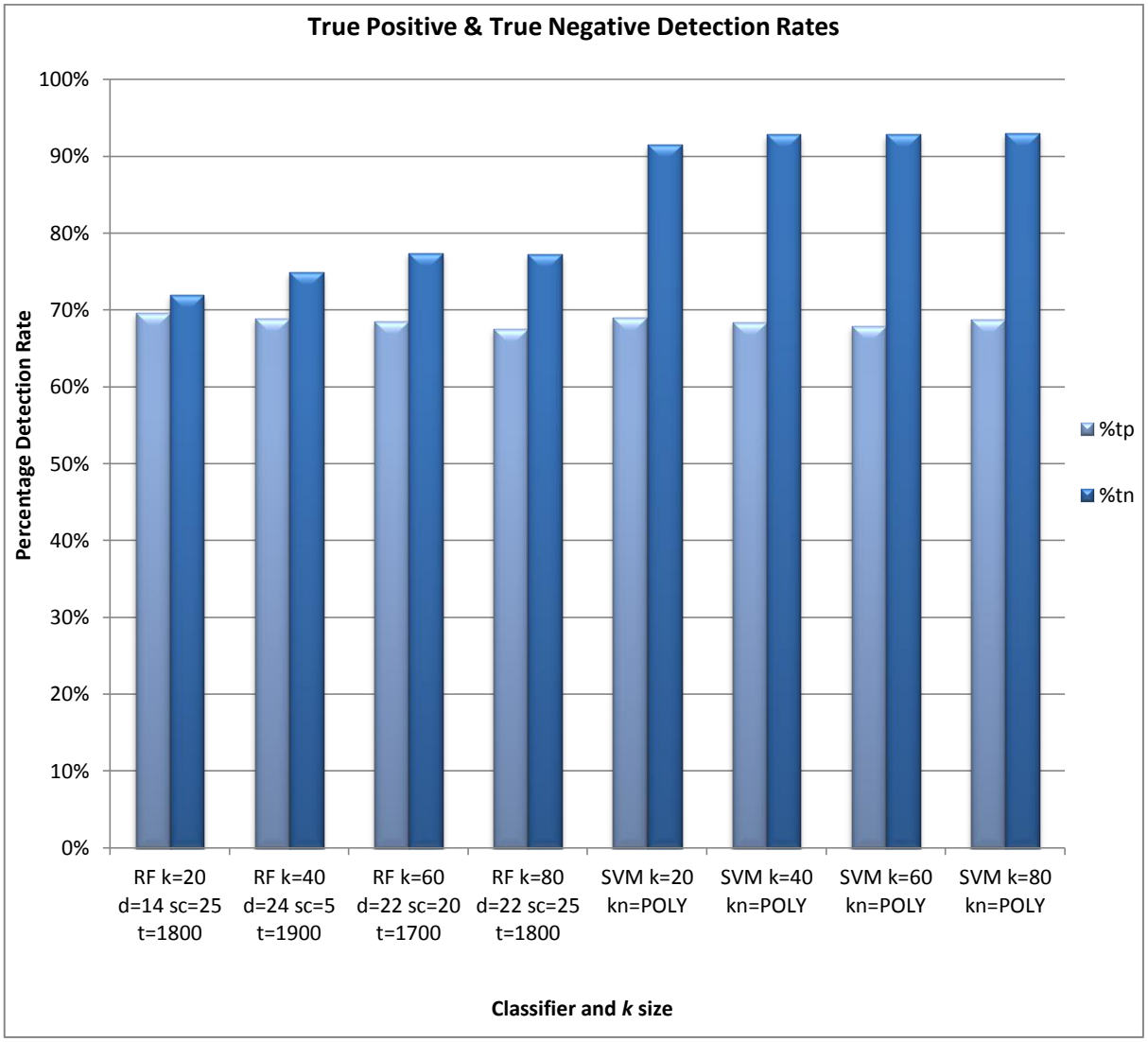


Graph 7 - True positive and true negative experimental results with raindrop context isolation

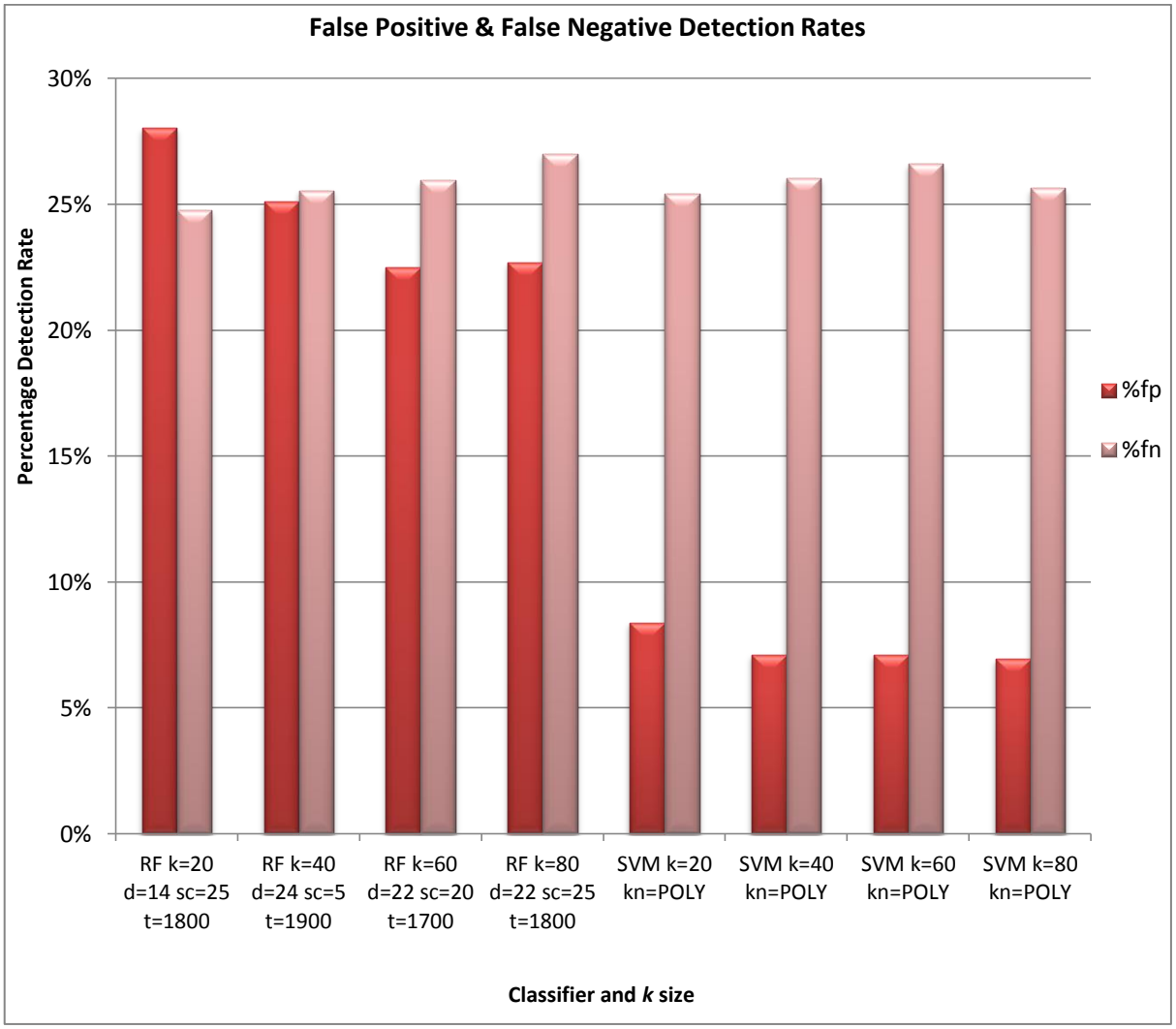
Graph 8 - False positive and false negative experimental results with raindrop context isolation



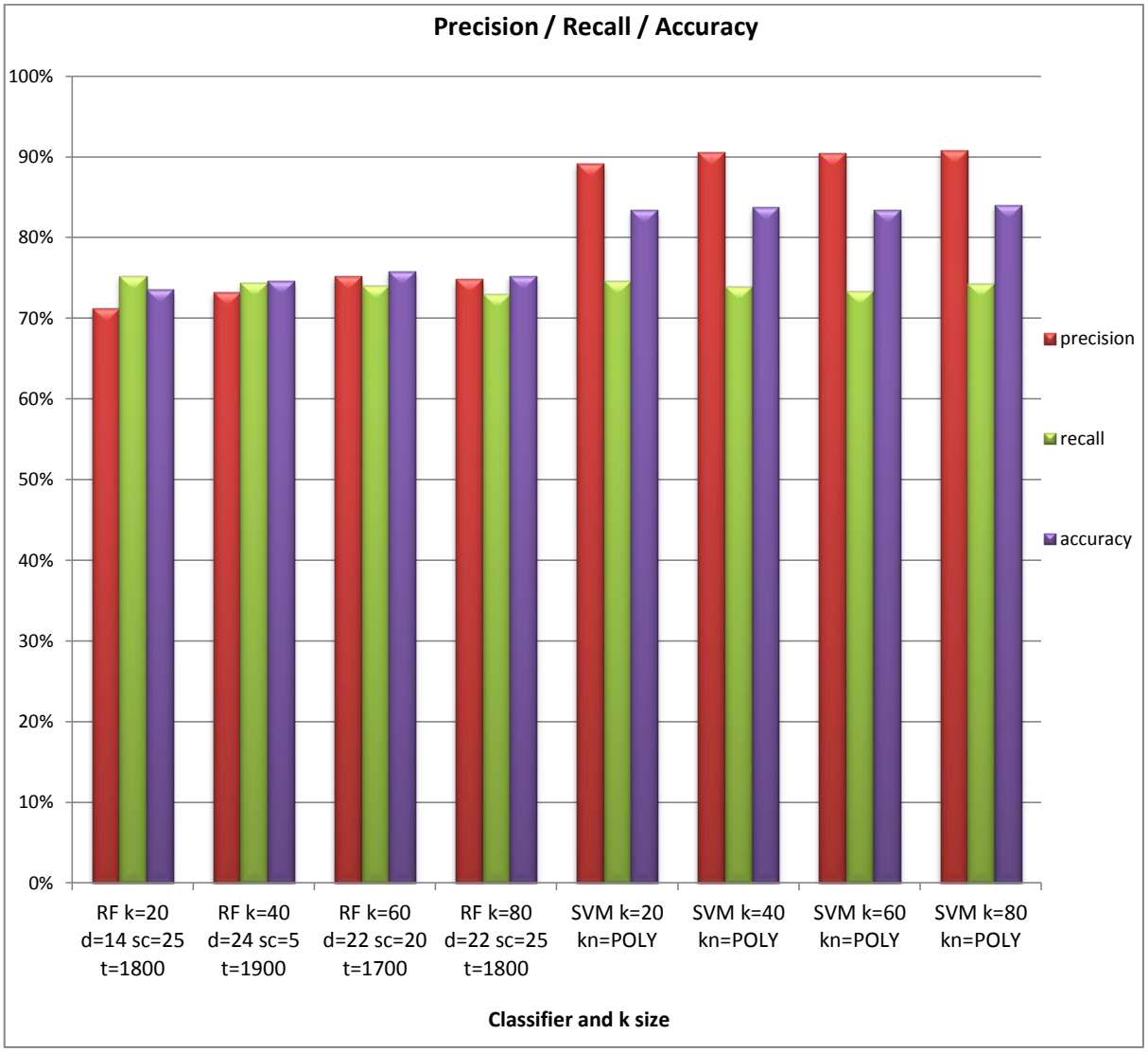
Graph 9 – Precision / Recall / Accuracy experimental results with raindrop context isolation



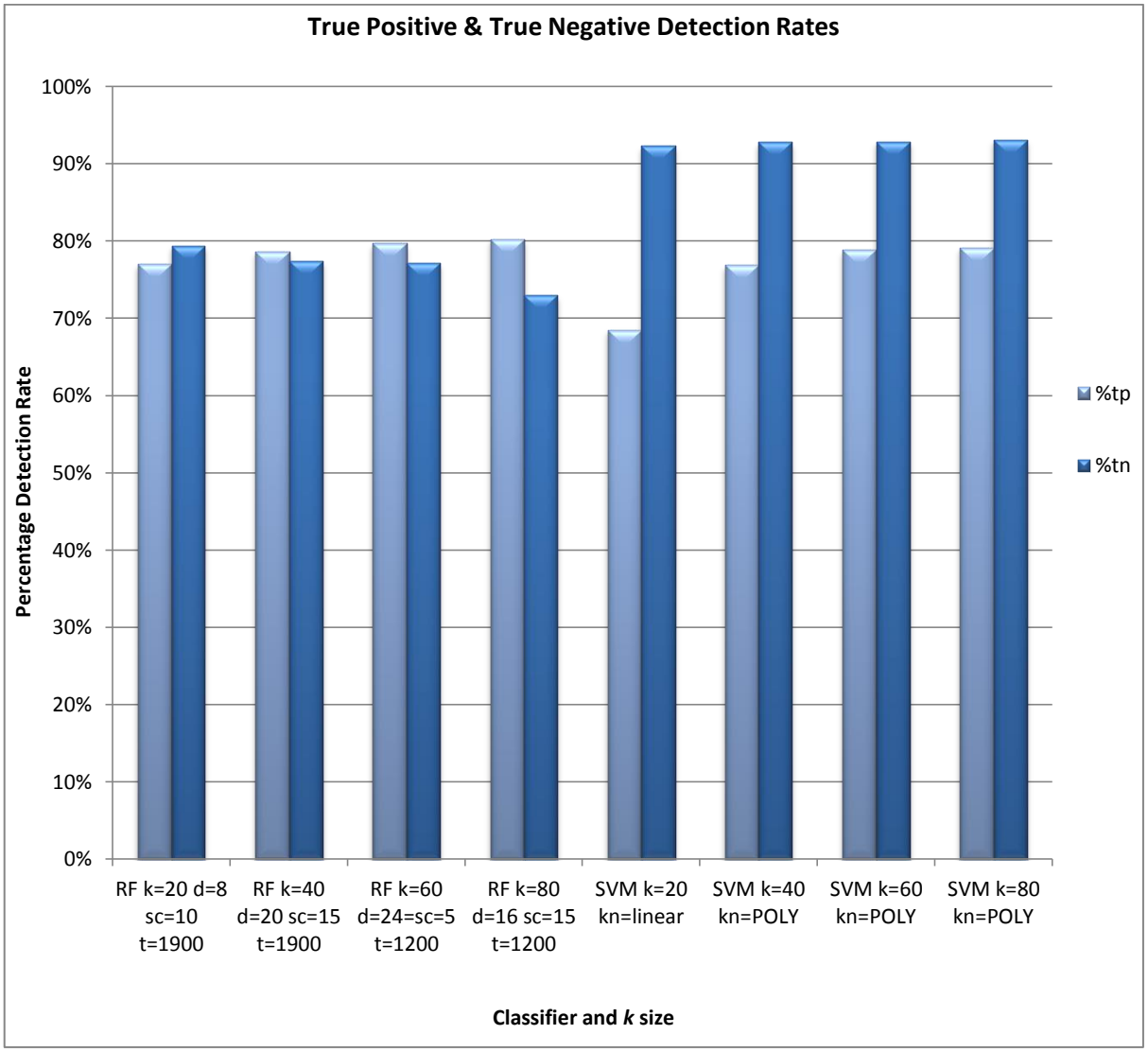
Graph 10 - True positive and true negative experimental results with Hu Moments



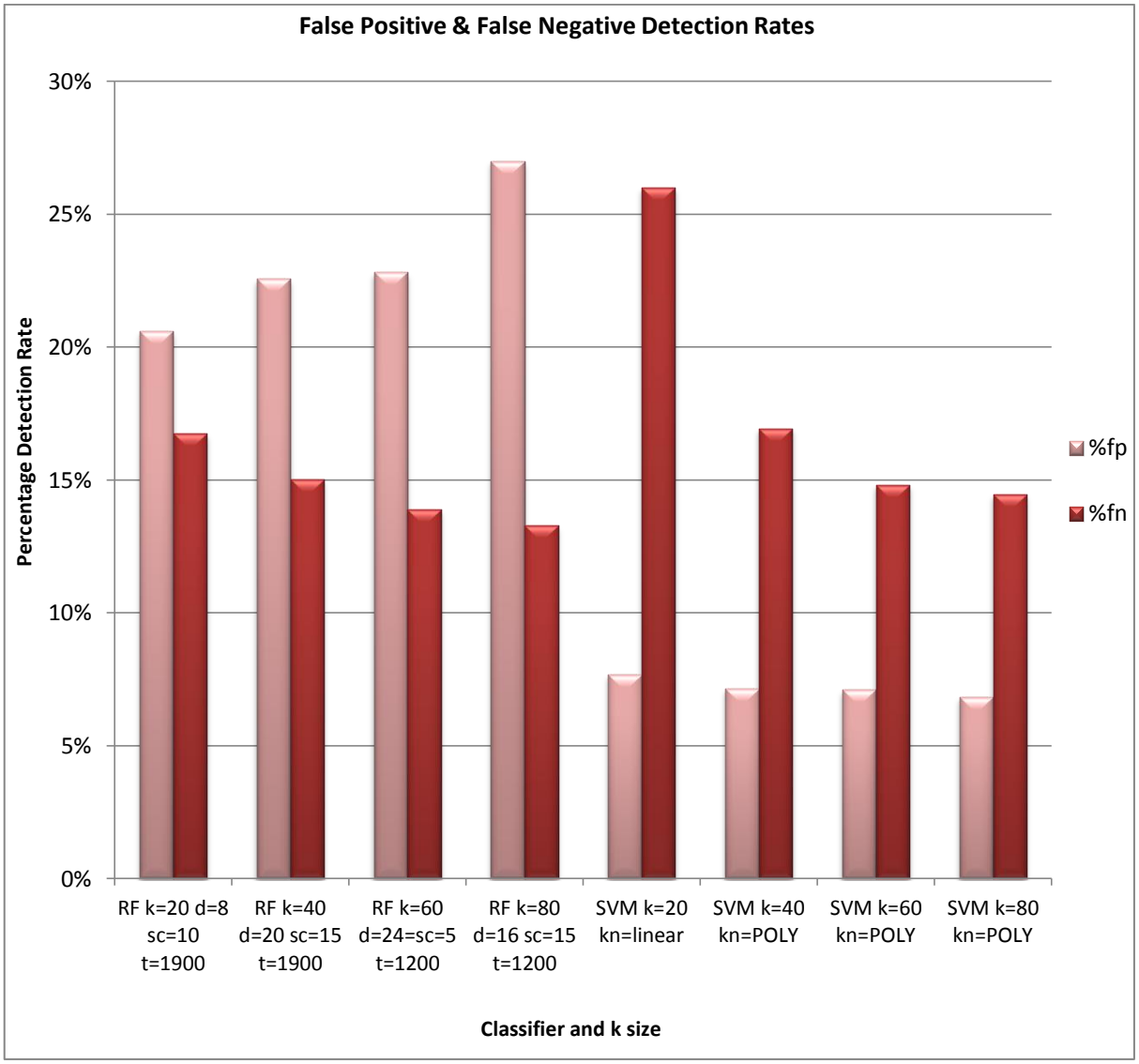
Graph 11 - False positive and false negative experimental results with Hu Moments



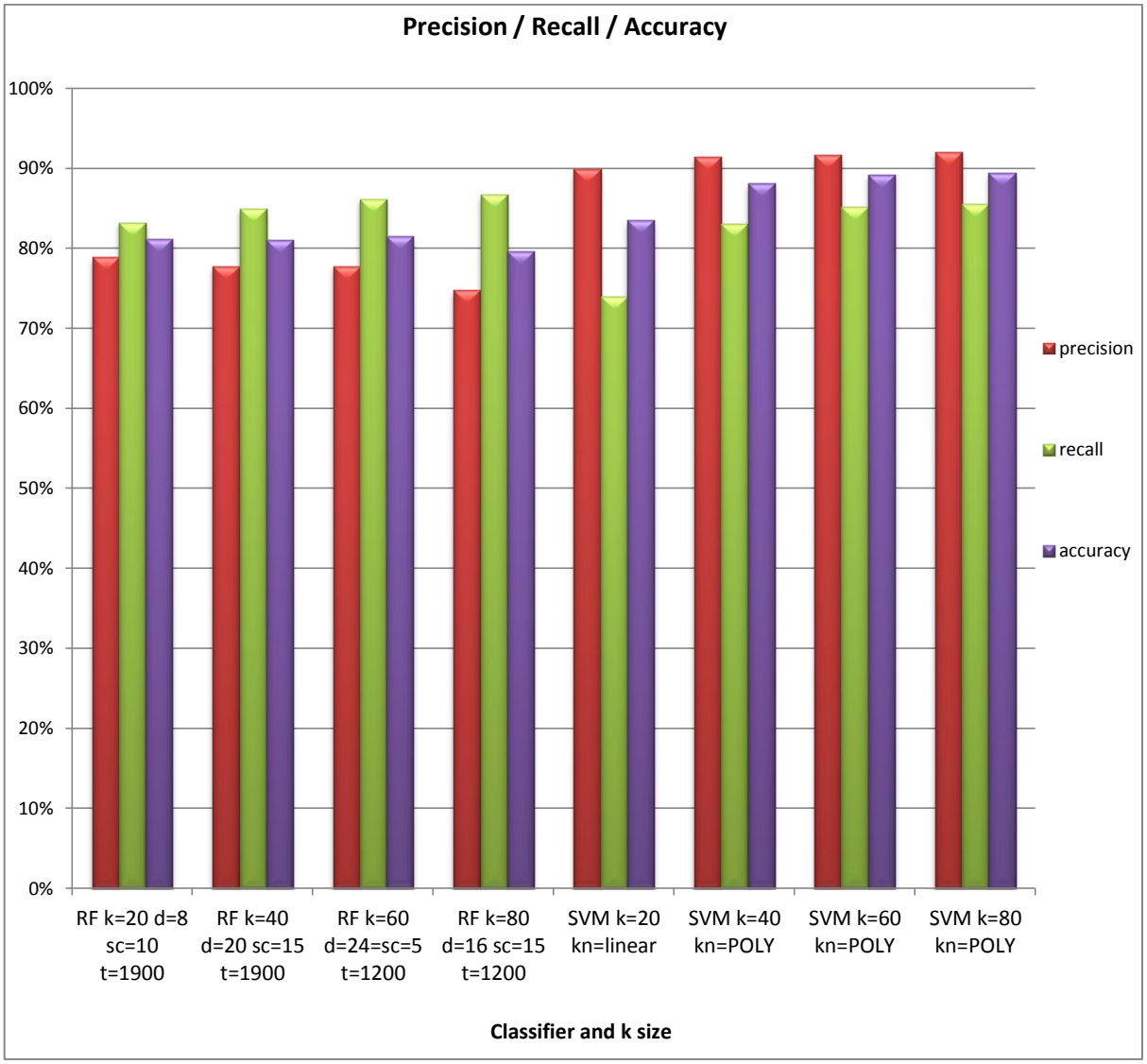
Graph 12 – Precision / Recall / Accuracy experimental results with Hu Moments



Graph 13 - True positive and true negative experimental results with raindrop context isolation and Hu Moments



Graph 14 – False positive and false negative experimental results with raindrop context isolation and Hu Moments



Graph 15 – Precision / Recall / Accuracy experimental results with raindrop context isolation and Hu Moments

10 REFERENCES

[1] Q. Wu, W. Zhang, and B. V. K. Vijaya Kumar, “Raindrop detection and removal using salient visual features,” in *IEEE Int.Conf.*, 2012, pp. 941–944.

[2] M. L. Eichner and T. P. Breckon, “Integrated Speed Limit Detection and Recognition from Real-Time Video,” in *IEEE Intell. Vehi.Symp.*, 2008, pp. 626–631.

[3] M. Kowaliszyn and T. P. Breckon, “Automatic road feature detection and correlation for the correction of consumer satellite navigation system mapping,” *IET Road Trans.Info. Control Conf. ITS UK Members’ Conf. Better Transp. through tech.*, pp. 02–02, 2010.

[4] M. L. Eichner and T. P. Breckon, “Real-Time Video Analysis for Vehicle Lights Detection using Temporal Information,” in *4th Euro. Conf. on Visual Media Prod.*, 2007, pp. I–9.

[5] A. Kheyrollahi and T. P. Breckon, “Automatic Real-time Road Marking Recognition Using a Feature Driven Approach,” *Mach. Vis. Appl.*, vol. 23, no. 1, pp. 123–133, 2012.

[6] T. P. Breckon, A. Gaszczak, J. Han, M. L. Eichner, and S. E. Barnes, “Multi-Modal Target Detection for Autonomous Wide Area Search and Surveillance,” in *SPIE Emerging Technologies in Sec. and Def.: Unmanned Sensor Sys.*, 2013, vol. 8899, pp. 1–19.

[7] T. P. Breckon, J. Han, and J. Richardson, “Consistency in Muti-modal Automated Target Detection using Temporally Filtered Reporting,” in *SPIE Electro-Optical Remote Sensing, Photonic Tech. and Appl. VI*, 2012, vol. 8542, no. 85420L-1, pp. 23:1–23:12.

[8] K. Garg and S. K. Nayar, “Detection and removal of rain from videos,” in *Comp.Vis. and Patt.Rec, IEEE Comp. Soc.*, 2004, vol. 1, pp. I–528–I–535.

[9] J. C. Halimeh and M. Roser, “Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation,” in *IEEE Intel. Vehi. Symp.*, 2009, pp. 610–615.

[10] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[11] M.-K. Hu, “Visual pattern recognition by moment invariants,” *IRE Trans. Inf. Theory*, vol. 8, no. 2, pp. 179–187, Feb. 1962.

[12] Y. Nomura and H. Naruse, “Reduction of obscuration noise using multiple images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 2, pp. 267–270, Mar. 1988.

[13] A. Yamashita, I. Fukuchi, and T. Kaneko, “Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information,” *IEEE Int. Conf.*, pp. 3794–3801, Oct. 2009.

[14] K. Garg and S. K. Nayar, “When does a camera see rain?,” *IEEE Int. Conf.*, pp. 1067–1074, 2005.

[15] A. Yamashita, T. Harada, T. Kaneko, and K. T. Miura, “Removal of adherent noises from images of dynamic scenes by using a pan-tilt camera,” in *IEEE Int. Conf.*, 2004, vol. 1, pp. 437–442 vol.1.

[16] A. Yamashita, T. Kaneko, and K. T. Miura, “A virtual wiper-restoration of deteriorated images by using a pan-tilt camera,” in *IEEE Int. Conf.*, 2004, pp. 4724–4729.

[17] M. Roser and A. Geiger, “Video-based raindrop detection for improved image registration,” *IEEE Int. Conf.*, pp. 570–577, Sep. 2009.

[18] W.-J. Park and K.-H. Lee, “Rain Removal Using Kalman Filter in Video,” *Int.Conf. on Smart Manuf. Appl.*, pp. 494–497, Apr. 2008.

[19] M. Shen and P. Xue, “A fast algorithm for rain detection and removal from videos,” *Multimed. Expo (ICME), 2011 IEEE …*, pp. 1–6, 2011.

[20] M. Bertalmio and L. Vese, “Simultaneous structure and texture image inpainting.,” *IEEE trans. image proc. a pub. IEEE Sig.Proc. Soc.*, vol. 12, no. 8, pp. 882–9, Jan. 2003.

[21] a. Yamashita, Y. Tanaka, and T. Kaneko, “Removal of adherent waterdrops from images acquired with stereo camera,” *IEEE Int. Conf.*, pp. 400–405, 2005.

[22] A. Yamashitat, M. Kuramotot, T. Kaneko, and K. T. Miurat, “A virtual wiper - Restoration of Deteriorated Images by Using Multiple Cameras,” in *IEEE Int. Conf.*, 2003, no. October.

[23] S. You, R. T. Tan, R. Kawakami, and K. Ikeuchi, “Adherent Raindrop Detection and Removal in Video,” in *IEEE Int. Conf. on Comp.Vis. and Pat.Recog.*, 2013, pp. 1035–1042.

[24] Y. Wexler, E. Shechtman, and M. Irani, “Space-time video completion,” *IEEE Comp.Soc.Conf. Comp.Vis. Pat.Recog.*, vol. 1, pp. 120–127.

[25] J. C. Halimeh and M. Roser, “Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation,” *Intell. Veh. Symp.*, pp. 610–615, 2009.

[26] R. Szeliski, *Computer vision: algorithms and applications*, 1st ed. New York: Springer-Verlag New York, Inc., 2010.

[27] J. J. Verbeek, N. Vlassis, and B. Kröse, “Efficient greedy learning of Gaussian mixture models,” *Neural Comput.*, vol. 15, pp. 469–485, 2003.

[28] S. Gormer, A. Kummert, S.-B. Park, and P. Egbert, “Vision-based rain sensing with an in-vehicle camera,” in *Intell. Veh. Symp.*, 2009, pp. 279–284.

[29] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, 1st ed. Prentice Hall Professional Technical Reference, 2002.

[30] C. J. Solomon and T. P. Breckon, *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*, 1st ed. Wiley-Blackwell, 2010.

[31] A. Treisman and G. Gelade, “A feature-integration theory of attention,” *Cogn. Psychol.*, vol. 136, pp. 97–136, 1980.

[32] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg, “The representation of visual salience in monkey parietal cortex.,” *Nature*, vol. 391, no. 6666, pp. 481–4, Jan. 1998.

[33] R. Desimone and J. Duncan, “Neural mechanisms of selective visual attention.,” *Annual review of neuroscience*, vol. 18. pp. 193–222, Jan-1995.

[34] L. Itti and C. Koch, “Computational modelling of visual attention.,” *Nat. Rev. Neurosci.*, vol. 2, no. 3, pp. 194–203, Mar. 2001.

[35] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *Pattern Anal. Mach. Intell.*, vol. 20, pp. 1254–1259, 1998.

[36] M. Cerf and J. Harel, “Predicting human gaze using low-level saliency combined with face detection,” *Adv. Neural Inf. Process. Syst. 20*, pp. 241–248, 2007.

[37] A. Yamashita, I. Fukuchi, T. Kaneko, and K. T. Miura, “Removal of adherent noises from image sequences by spatio-temporal image processing,” in *IEEE Int. Conf. Robotics and Automation*, 2008, pp. 2386–2391.

[38] P. E. Duda, Richard O. and Hart, “Use of the Hough Transformation to Detect Lines and Curves in Pictures,” *Commun. ACM*, vol. 15, no. April 1971, pp. 11–15, 1972.

[39] P. D. Wellner, “Adaptive thresholding for the DigitalDesk,” *Xerox, EPC1993-110*, 1993.

[40] T. Joachims, “Text categorization with Support Vector Machines: Learning with many relevant features,” *Mach. Learn. ECML-98 SE - 19*, vol. 1398, pp. 137–142, 1998.

[41] I. Pilászy, “Text Categorization and Support Vector Machines.” .

[42] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” in *In Workshop on Stat.Learning in Comp.Vis.*, 2004, pp. 1–22.

[43] D. Turcsany, A. Mouton, and T. P. Breckon, “Improving Feature-based Object Recognition for X-ray Baggage Security Screening using Primed Visual Words,” in *Inter. Conf. on Indust.Tech.*, 2013, pp. 1140–1145.

[44] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” in *IEEE Int. Conf. on Comp.Vis.*, 2003, pp. 1470–1477 vol.2.

[45] S. Suzuki and K. Be, “Topological structural analysis of digitized binary images by border following,” *Comput. Vision, Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.

[46] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[47] OpenCV Dev Team, “Introduction to Support Vector Machines,” 2014. [Online]. Available: http://docs.opencv.org/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. [Accessed: 06-May-2014].

[48] U. A. udiprod Aharoni, “SVM with polynomial kernel visualization,” 2007. [Online]. Available: http://www.youtube.com/watch?v=3liCbRZPrZA. [Accessed: 06-Jun-201AD].

[49] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

[50] D. Benyamin, “A Gentle Introduction to Random Forests, Ensembles, and Performance Metrics in a Commercial System,” 2012. .

[51] S. Klupsch and M. Ernst, “Real Time Image Processing based on Reconfigurable Hardware Acceleration,” no. 0, pp. 1–7.

[52] Altera, “White Paper Video and Image Processing Design Using FPGAs,” 2007.

[53] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[54] N. Fraser, “Neural Network Follies,” 1998. [Online]. Available: https://neil.fraser.name/writing/tank/. [Accessed: 23-Mar-2014].